SELF-TUNED NEURO-FUZZY CONTROLLER BASED INDUCTION MOTOR DRIVE

By

Zhi Rui Huang

A thesis submitted in partial fulfilment of the requirement for the of Masters of Science

at

Lakehead University

Thunder Bay, Ontario

August 2007

©Copyright by Zhi Rui Huang, 2007

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-33589-5
Our file Notre référence
ISBN: 978-0-494-33589-5

NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:
L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Abstract

Among various ac motors, induction motor (IM) occupies almost 90% of the industrial drives due to its simple, robust construction and generally satisfactory efficiency as compared to dc motor. However, the control of IM is complex due to its nonlinear nature and the parameters change with operating conditions. Since 1980s, field orientation principle (FOP) has been used for high performance control of IM. Due to the well-known drawbacks of the fixed-gain proportional-integral (PI), proportional-integral-derivative (PID) and various adaptive controllers, over the last two decades researchers have been working to apply artificial intelligent controller (AIC) for IM drives due to its advantages as compared to the conventional PI, PID and adaptive controllers. The main advantages are that these controllers can handle any nonlinearity of arbitrary complexity, and their performances are robust. Also fuzzy rules and neural network (NN) can be used to model a process for model reference or model predictive control. Meanwhile, the designs of these controllers do not depend on accurate system mathematical model. Neuro-fuzzy controller (NFC), as a kind of artificial intelligent controller (AIC), has attracted much attention by researchers as it takes advantages from both fuzzy logic controller (FLC) and NN by combining the expert human knowledge and the learning ability of the NN.

Despite lots of research on AIC application for motor drives, industries are still reluctant to use AIC for real-life industrial drives. The main reason is that most of AIC require complex calculation and hence suffer from high computational burden. Therefore, attention needs to be paid to develop AIC which is suitable for practical

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

applications. In order to achieve that, in this thesis, first, a novel, low computational and simplified self-tuned NFC is developed for the speed control of IM drive. For the proposed NFC only the speed error is used as the input, unlike conventional NFCs, which utilize both speed error and its derivative as inputs. Obviously, this simplification lowers down computational burden and makes the NFC easier to be implemented in practical applications. Next, a faulty IM with broken rotor bars (IMBRB) is considered and a NFC is developed to minimize the speed ripple of that motor. The speed error and rotor electrical angle are used as two inputs of the NFC.

A supervised self-tuning method is also developed for the developed NFCs. The system error, instead of controller error, has been utilized to tune the membership functions and weights because the desired controller output is not readily available. Also the convergences/divergences of the weights are analyzed and investigated.

Simulation models for indirect field oriented control of IM incorporating both of the developed NFCs are developed in Matlab/Simulink. IM drives based on both of the developed NFCs are successfully implemented in real-time using DSP board DS1104. For the first NFC, comparisons with conventional NFC and PI are done both in simulation and experiment at different operating conditions for a laboratory 1/3 hp IM. Also the effectiveness of the second NFC is tested for a laboratory 0.5 hp IMBRB both in simulation and experiment, compared to a well-tuned PI controller. It is found from the experimental results that the proposed NFC reduces the fundamental and second harmonic components of speed ripple which are significant components as compared to high frequency components.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Acknowledgements

I would like to express my most sincere thanks and appreciation to my supervisor, Dr. M. Nasir Uddin and my co-supervisor, Dr. Wilson Wang for their valuable encouragement and guidance step by step throughout the program.

I also thank my colleagues, graduate student Md Muminul Chy, Xijiang Dou, Nairui Yin, Ming Gong, Wenguang Li for the inspirational and pleasant discussions on academic topics.

Finally, I would express my sincere love to my wife, He Huang for her continued support and patience during the last two years, and my baby girl, Sophia Huang, for the good luck she brings to me.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Table of Contents

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

List of Figures

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Chapter 1

Introduction

This chapter provides an introduction about induction motor (IM) and a review of the state-of-the-art-work on the control of the IM drivers. At the end the motivation and organization of this work are also provided.

1.1 Structure of Induction Motors

The IM have been used as a workhorse in the industry due to its simple and robust construction. The *squirrel-cage motor* is a type of widely used IM, as compared to the *wound-rotor IM*. The squirrel-cage motor has unwired and inaccessible rotor and owns the additional degree of ruggedness. Therefore, only squirrel-cage induction motors are considered in this thesis.

In general, an IM consists of three basic components as shown in Fig. 1.1.

1) *Stator*: Houses the stator core and windings. The stator core consists of many layers of laminated steel, which is used as a medium for developing magnetic fields.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

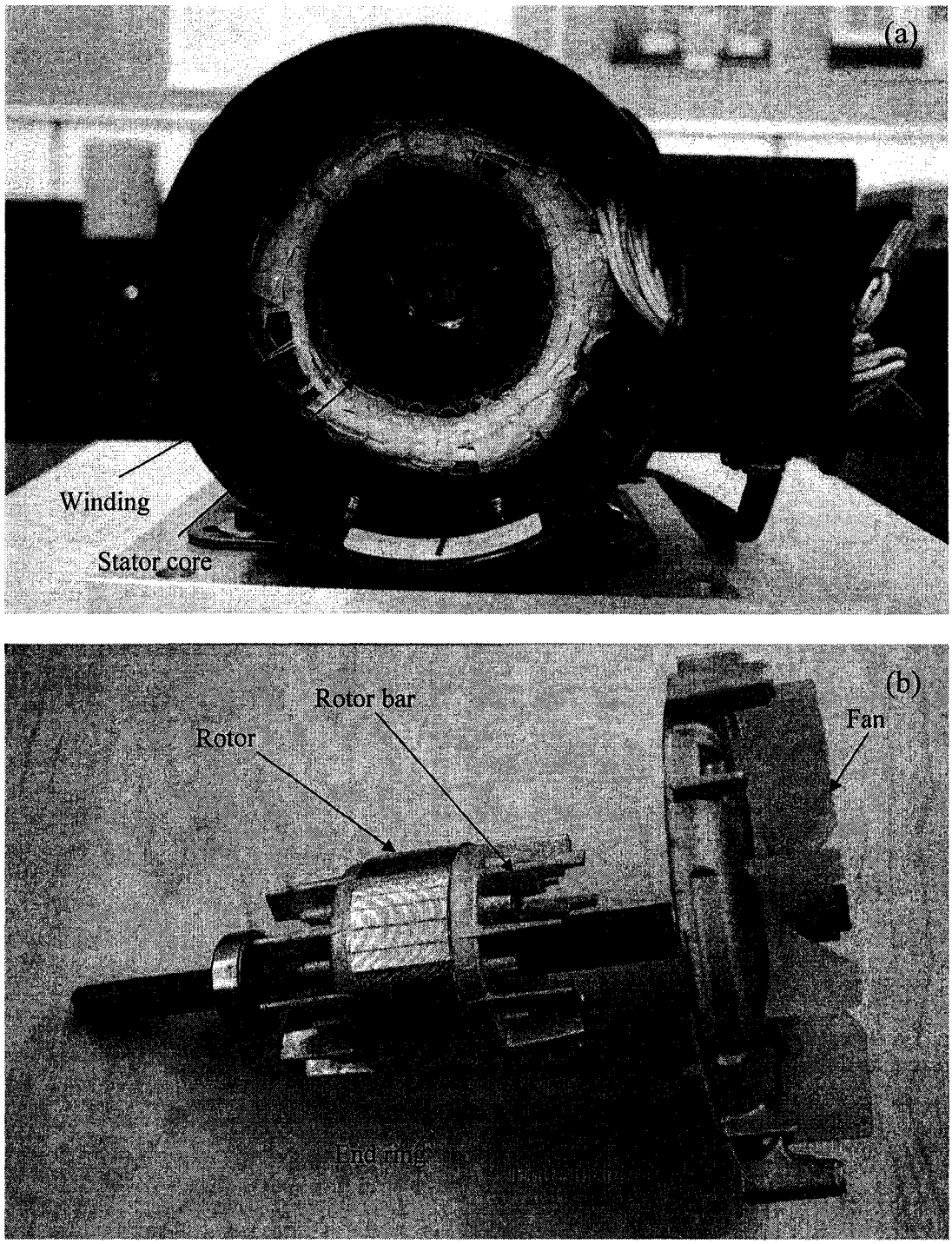


Fig. 1.1 Structure of squirrel-cage induction motor: (a) stator, (b) rotor.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

2) *Rotor*: Also constructed of many layers of laminated steel. The rotor windings consist of bars of copper or aluminium alloy shorted, at either end, with shorting rings.

3) *End shields*: Support the bearings which center the rotor within the stator.

The basic principle of operation of IMs is that the rotating magnetic field acts upon a rotor to develop mechanical torque. The stator winding of the IMs are evenly distributed by 120 electrical degrees. As the three-phase current enters the stator windings, it creates a rotating magnetic field within the air gap (the space between the stator and rotor). The speed that the fields rotate around the stator is known as the synchronous speed (N_S). As the magnetic field revolves, it cuts the conductors of the rotor and generates a current flowing in the rotor conductors. This creates another magnetic field which interacts with the air gap field producing a torque. Consequently, the motor rotates at a speed $N < N_S$ in the direction of the rotating field. The actual output speed of the rotor is related to the synchronous speed and the slip, S, as:

$$S = \frac{N_S - N}{N_S}. \tag{1.1}$$

1.2 Literature Review

Nowadays IMs have been widely utilized in various industrial variable-speed drive applications because of the maintenance advantages and less expensive over dc motor drivers. However the control of IMs is still a challenging problem due to the following reasons:

1) The dynamical system is nonlinear.

2) Two of the state variables (rotor fluxes/currents) are not usually measurable.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

3) Due to heating, the rotor resistance varies considerably with a significant impact on the system dynamics [1].

The development of high-performance control theory for ac drives, driven by industry requirements, has followed a rapid evolution during the last two decades. One comprehensive process of IM control system design involves two levels: first to choose a ***control strategy***; and then to design a ***controller.***

As shown in Fig. 1.2 [1], IMs control strategies can be broadly classified into two categories such as ***scalar control*** and ***vector control.***

Although simple to be implemented, ***Scalar control*** has only been used in low-performance, cost-effective industry drives. In scalar control, only magnitude and frequency (angular speed) of voltage, current, and flux linkage space vectors are controlled. The most common scalar technique is that of constant *volts/frequency* (v/f). In v/f method magnitude of stator voltage is adjusted in proportion to the frequency in order to maintain an approximately constant stator flux in the IM. The v/f method consists of controlling the speed of the rotating magnetic field of the stator by changing the supply frequency. However, as the scalar control strategy is based on steady-state principles, the transient performance is not optimized. For example, when starting an IM directly on line, many times full load torque is developed with inrush currents of perhaps six times than rated one.

On the contrary, in ***vector control***, which is based on relations valid for dynamic states, not only magnitude and frequency (angular speed), but also instantaneous positions of voltage, current, and flux space vectors are controlled. Thus, the vector control acts on the positions of the space vectors and provides their correct orientation both in steady state and during transients. The most popular two

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

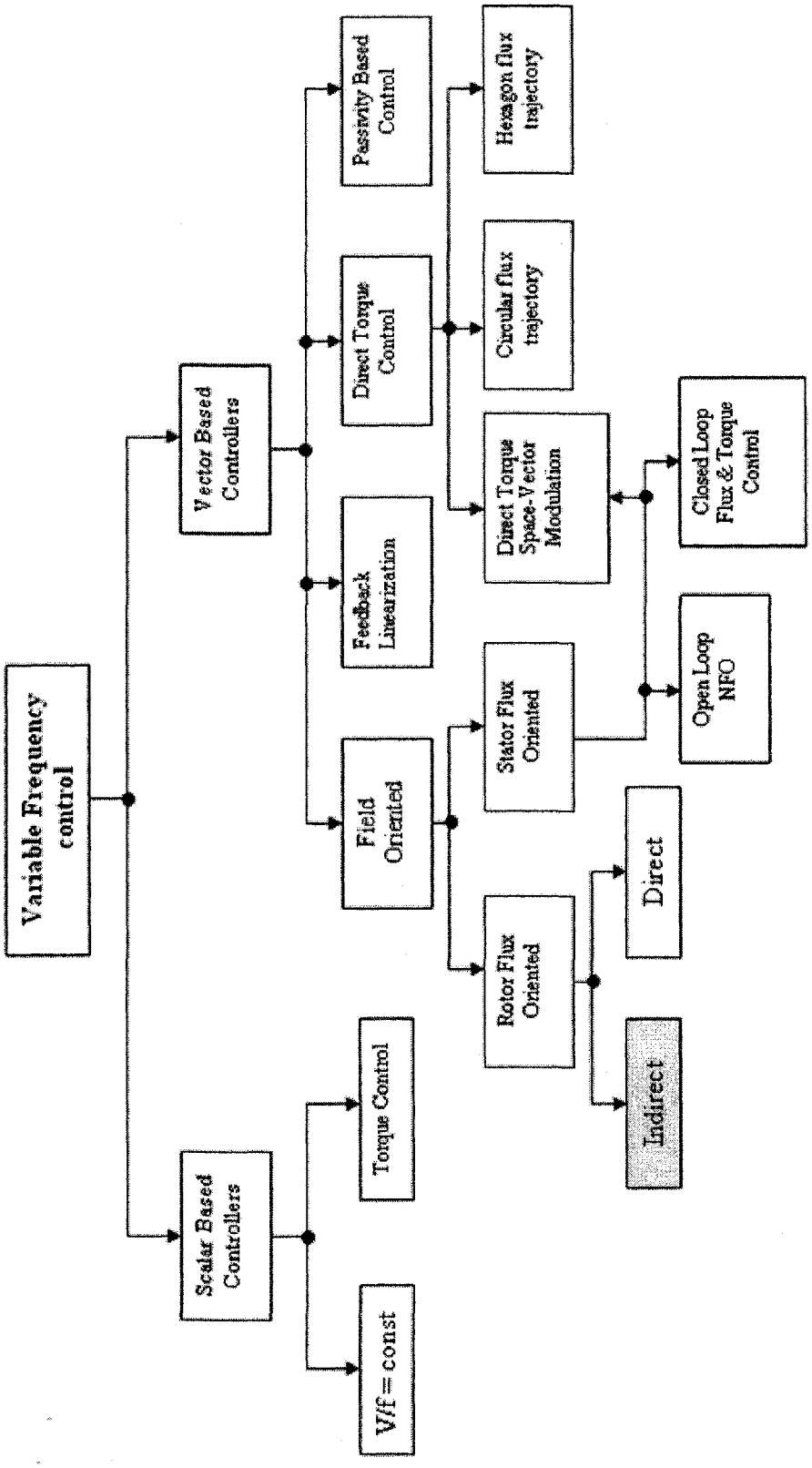


Fig. 1.2 Classification of IM control strategies.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

vector methods are known as *field-oriented control* (FOC) [2]-[4] and *direct torque control* (DTC) [5]-[9]. They have been invented respectively, in 70's and 80's. These control strategies are different on the operation principle but their objectives are the same. They both aim to control effectively the motor torque and flux in order to force the motor to accurately track the command trajectory regardless of the machine and load parameter variation or any extraneous disturbances. Both control strategies have been successfully implemented in industrial products.

There also exist other vector control methods implemented in different ways. Marino et al. [10] has proposed a nonlinear transformation of the motor state variables so that, in the new coordinates, the speed and rotor flux amplitude are decoupled by feedback. The method is called *feedback linearization control* (FLC) or *input–output decoupling* [5], [11]-[13]. A similar approach, derived from a *multi-scalar* model of the induction motor, has been proposed by Krzeminski [15]. A method based on the *variation* theory and energy shaping called *passivity-based control* (PBC) [14] has been investigated. In this case, an IM is described in terms of the Euler–Lagrange equations expressed in generalized coordinates.

Numerous controllers have been utilized in IM control based on control strategies mentioned before. The general classification of the controllers is presented in Fig. 1.3.

Over a long time, conventional linear controllers such as PI, PID have been widely applied to IM drives. But these controllers are sensitive to parameter variations and load disturbance. The performance depends on operating conditions and PI gains and it is also difficult to tune PI gains to solve the overshoot and load disturbance rejection problems simultaneously. Overshoot elimination setting will cause a poor load disturbance rejection, and rapid load disturbance rejection setting will cause

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

overshoot or even instability in the system. To overcome this problem, PI/PID controllers with tuning gains have been proposed [18]-[23]. However, the complicated calculation makes the tuning either be done off-line [18]-[21] or on-line with a very long sampling time such as 10ms [22] and 50ms [23] which are not acceptable in some applications.

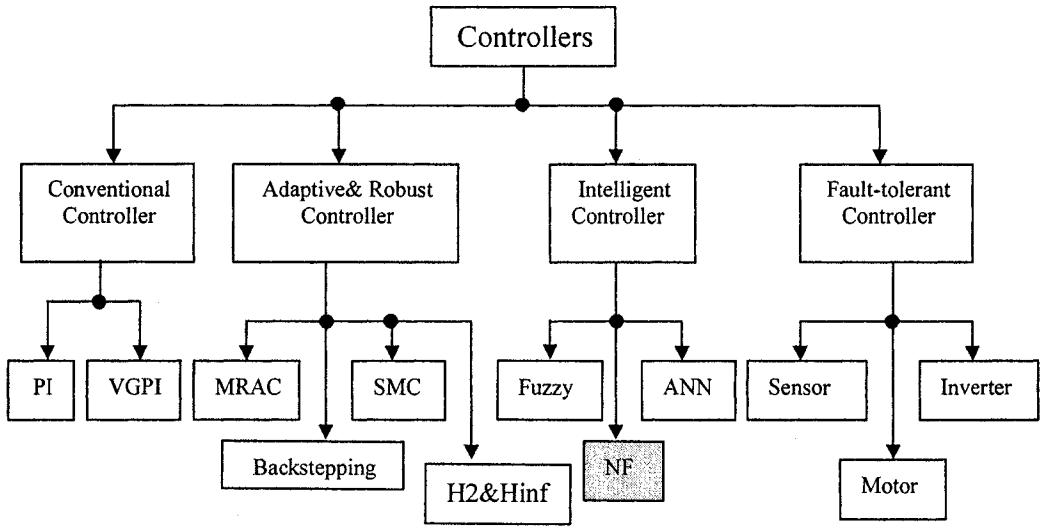


Fig. 1.3 Classification of controllers for IM.

In order to avoid the shortcomings of PI/PID controllers, researchers applied adaptive control techniques for IM drives to achieve parameter insensitivity and fast response [24]-[30]. Lorenz and Lawson [24] proposed the model reference adaptive control (MRAC). This adaptation functions by creating an error signal between a reference motor model and an estimated quantity based on motor outputs. This error will modify a gain in the system until the error is driven to zero. However, the utilization of the motor voltage terminal quantities increases the system complexity. Also these signals can be noisy and degrade the system performance. Rowan and Kerkman [25] presented some other models and found that none of them will provide

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

a total solution to the detuning problem. Applications of sliding mode control (SMC) in ac drives, mostly in position control systems, have been reported in [26]-[28]. The structure and design of SMA is relatively simple. However, the SMC is facing the chattering problem which puts unacceptable vibration stress on the load [28] due to the finite sample times of microprocessor implementation. The authors of [30] combined field orientation and adaptive backstepping approach for the control of IM. The design of the control law and the estimation rotor resistance and load torque were based on the nonlinear model of IM. However, it is often difficult and sometimes almost impossible to develop an accurate mathematical model of an IM.

$H2\&H\infty$ control theory is also used to design robust controllers for IM systems [31]-[36] . Field orientation or feed back linearization has to combine with $H2\&H\infty$ theory in order to eliminate IM intrinsic nonlinearity. The $H2\&H\infty$ controllers demonstrated robustness to parameters variations and exogenous disturbances, but are highly dependent on accurate system model and parameters. In addition complex calculation makes its application limited.

In order to overcome the disadvantages of conventional PI, PID and adaptive controllers, recently researcher applied intelligent controllers for motor driver applications. The main advantages of intelligent controllers are: the design of intelligent controllers is independent of the system parameters and it can handle system nonlinearity.

The artificial neural network (ANN) is well known for its learning ability and approximation to any arbitrary continuous function. The ANN also possesses the ability to perform in noisy environments and is tolerant to faults and missing data. Some work has been reported on the use of ANN controllers for IM drives [37]-[41]. However, due to the iterative nature of the neural network, training of the ANN is too

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

slow for certain applications. For example, the authors in [38] presented a FOC control scheme for IM incorporating an ANN controller with two input nodes, eight hidden log-sigmoid neurons and three pure linear output neurons. The ANN was trained with 750 samples with a 1% sum-squared error goal. The proposed ANN converged after 80,000 iterations. In [39], the authors proposed a simple network structure with 4 input nodes, 2 output nodes and no hidden layers for the vector control of IM. The error tolerant was set to be 0.1% of the command speed. The ANN was trained by an unsupervised algorithm and took "at least a few seconds" to reach the goal. Because of this reason, most of ANN controllers are trained off-line. However, for off-line training, the ANN controllers need a large amount of data in order to cover all the operating conditions. The ANN controller exhibits unacceptably poor performance outside of the operating conditions over which they have been trained [37]. Also it is difficult for an off-line-trained ANN controller to cope with the dramatically changed environment or system parameters. On the other hand the on-line training algorithms generally take too much computational overhead and limit the sample frequency of the overall system [40]. In order to reduce the execution time of the ANN controller, authors in [41] proposed a network structure with partial fixed-weights. But the values of these fixed-weights depended on accurate mathematical models which were not always available.

The theory of fuzzy-logic controller (FLC) is based on the linguistic rules with an IF-THEN general structure, which is the basis of human logic. The FLC could also handle the nonlinearity of arbitrary complexity. These advantages make FLC attractive to IM drives [43]-[48]. However, the design of FLC depends on the expertise and trial and error procedure. The values of the constants, membership functions, fuzzy sets for the input/output variables, and the rules are all need to be

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

adjusted by trial and error if optimized performance is wanted [43]. To make the things worse, a fuzzy controller implemented in the motor drive speed control usually has asymmetric membership functions [43], [44]. It is a time-consuming job to find proper parameters for these membership functions even by an expert.

Neuro-Fuzzy Controller (NFC) has attracted much attention due to its combined ideas FLC and ANN. A NFC offers the control system designer the opportunity to make use of the advantages from both FLC and NN by combining transparent and linguistic control rules of FLC and the learning ability of the NN. The NFC has been utilized by researchers for motor drive applications [49]-[59] and fault tolerant control of motor [60], [61]. A typical NFC has a four layer structure: input, fuzzfication, rules, and defuzzfication. This leads to the higher complexity compared to the FLC and ANN controller. The high complexity causes high computational overhead. Despite many advantages of NFCs, the industry has been still reluctant to apply these controllers for commercial drives due to high computational burden caused by large number of membership functions, weights and rules, especially on self-tuning condition [56], [57]. High computation burden leads to low sampling frequency, which is not sufficient for real-time implementation. In [50] the authors presented a four-layer NFC with 2 inputs, 3 membership functions each, 9 rules. Only rules (weights) were tuned to lower the computational burden, but the cost is that the parameters of membership functions have to be chosen by trial and error procedure. In [58] the authors proposed a five-layer NFC: input, fuzzfication, rules, normalization, and defuzzfication, in the discrete direct torque control of IM scheme. The authors found relatively high torque ripple caused by low sample rate.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

1.3 Thesis Motivation

In this work the indirect field-oriented control (IFOC) is selected as a control strategy because of its simple structure and low implementation complexity.

As discussed in literature review, the high complexity and computational burden have prevented the practical utilization of the NFCs. Thus in this thesis a low computational self-tuned NFC is developed for the speed control of IM drive. The objective of this work is to simplify the complicated structure of the conventional NFCs and at the same time maintain the system performance to the most extent.

Next, the work has been extended for a faulty IMBRB which suffers from high frequency speed ripple and hence mechanical vibration. Conventionally, first, the fault is diagnosed and then a fault tolerant controller (FTC) is developed to minimize the speed ripple. However, the development of a model for the fault and hence a FTC is difficult and time consuming. Therefore, in this thesis, first, the speed ripple of IMBRB is investigated through an open loop experiment. And then a NFC is developed to minimize the speed ripple of IMBRB. Thus, the developed NFC works as a type of FTC.

1.4 Thesis Organization

The remaining chapters of the thesis are organized as follows. In Chapter 2 the basic concept of field oriented control (FOC) and neuro-fuzzy controller (NFC) with fuzzy singleton rules are explained. The parameters tuning of NFC in a control system is also discussed in this chapter. Next, Chapter 3 shows the design procedure of the low computational neuro-fuzzy speed controller for an IM drive. Experimental setup and detail implementation procedure are described in this chapter. Simulation and

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

experimental results are shown. A supervised self-tuning method is also developed for the NFC. The system error, instead of controller error, has been utilized to tune the weights because the desired controller output is not readily available. Chapter 4 describes research work on the utilization of NFC for IMBRB and analyzes the convergences/divergence of the weights of the developed NFC. Finally, Chapter 5 presents a summary of the contributions of this work, future work and the conclusions. After that all pertinent references and appendices are listed.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Chapter 2

Field Orientation & Neuro-Fuzzy Control Techniques

This chapter introduces the theory of field orientation control (FOC) and the Neuro-fuzzy controller (NFC) with fuzzy singleton rules. The training methods of NFC are outlined in this chapter. The difference between the Mamdani's fuzzy and Sugeno-type fuzzy interferences is also explained. Also the problem related to application of NFC in the control systems is illustrated.

2.1 IM Model for FOC

FOC provides a method of decoupling the two components of stator currents: one producing the air gap flux and the other producing the torque. Therefore, independent control of torque and flux, which is similar to a separately excited dc motor, can be achieved. The magnitude and phase of the stator currents are controlled in such a way that flux and torque components of current remain decoupled during

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

transient and steady-state conditions. The FOC consists of three steps as follows,

Step1: Clarke's Transformation (*abc-dq*),

Step2: Park's Transformation(*dq-DQ*),

Step3: Rotor Flux Alignment.

2.1.1 Clarke's Transformation

Krause and Thomas [16] introduced a two phase equivalent machine representation of 3-phase IM. The changes of variables, so-called Clarke's transformation or $abc-\alpha\beta$, which transform the motor equations in the original three phase system to the equivalent two phase reference frame, for example, stator coils three phase voltages Vas, Vbs, Vcs into corresponding vectors in the stator reference frame, are given by,

$$\begin{bmatrix} u^s_{\alpha s} \\ u^s_{\beta s} \end{bmatrix} = \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} V_{as} \\ V_{bs} \\ V_{cs} \end{bmatrix} \tag{2.1}$$

The inverse Clarke's transformation, $abc-\alpha\beta$, can be performed as

$$\begin{bmatrix} V_{as} \\ V_{bs} \\ V_{cs} \end{bmatrix} = \begin{bmatrix} \frac{2}{3} & 0 \\ -\frac{1}{3} & \frac{1}{\sqrt{3}} \\ -\frac{1}{3} & -\frac{1}{3} \end{bmatrix} \begin{bmatrix} u^s_{\alpha s} \\ u^s_{\beta s} \end{bmatrix}. \tag{2.2}$$

Analogous transformations may apply to the other vector quantities of the IM. Thus the IM dynamic equations can be described in $\alpha-\beta$ coordinates as,

$$R_s i^s_{\alpha s} + \frac{\partial \lambda^s_{\alpha s}}{\partial t} = u^s_{\alpha s}, \tag{2.3}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$R_s i_{\beta s}^s + \frac{\partial \lambda_{\beta s}^s}{\partial t} = u_{\beta s}^s, \tag{2.4}$$

$$R_r i_{\alpha r}^r + \frac{\partial \lambda_{\alpha r}^r}{\partial t} = 0, \tag{2.5}$$

$$R_r i_{\beta r}^r + \frac{\partial \lambda_{\beta r}^r}{\partial t} = 0, \tag{2.6}$$

where R, i, λ, u denote resistance, current, flux linkage, and stator voltage input to the machine; subscripts *"s"* and *"r"* stand for stator and rotor, (α, β) denote the equivalent two phase axis, superscript *"s"* denotes the components of a vector with respect to a fixed stator reference frame, *"r"* denotes the components of a vector with respect to frame rotating at speed ω_r.

It is noted that the stator vector i_s^s, λ_s^s and u_s^s are revolving in the fixed stator reference frame with speed ω_e, and that the rotor vector i_r^r and λ_r^r are revolving with speed ω_{sl} in the rotor reference frame which is rotating at speed ω_r. And we have

$$\omega_e = \omega_r + \omega_{sl} \tag{2.7}$$

Figure 2.1 illustrates the relationship between stator and rotor reference frames.

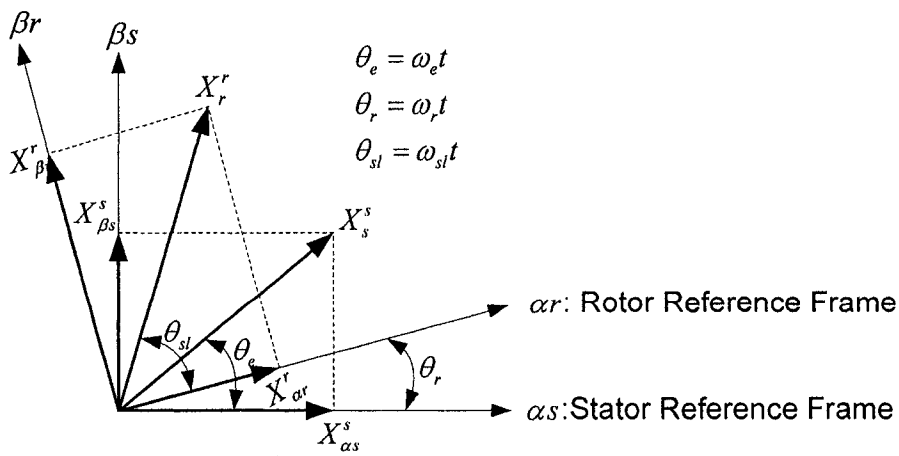


Fig. 2.1 Illustration of rotor reference frame and stator reference frame.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

2.1.2 Park's Transformation

From Fig. 2.1, it is easy to see that the vector quantities in $\alpha-\beta$ reference frame have sinusoidal ac waveforms. AC quantities are somewhat inconvenient as they change with rotor position. Therefore, another transformation, named Park's transformation, is introduced which allows conversion of the stationary $\alpha-\beta$ components of the motor vectors into synchronously rotating $\alpha-\beta$ frame where the quantities become fixed. The Park's transformation involves the excitation reference frame (d,q) which rotates with the speed ω_e as shown in Fig. 2.2.

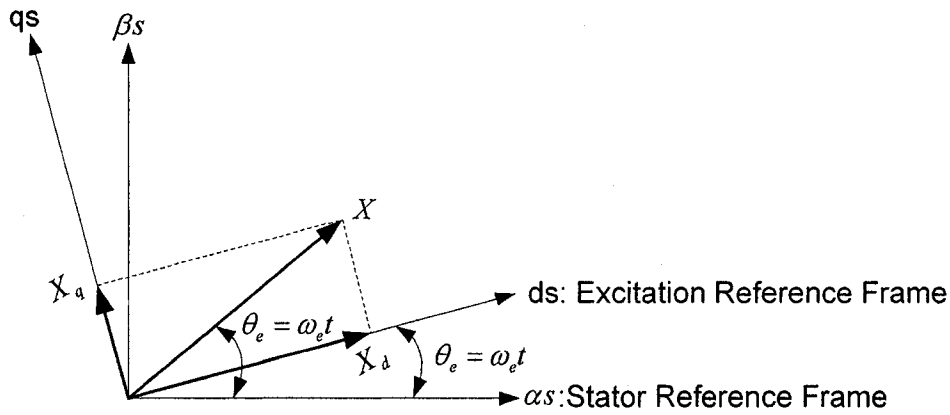


Fig. 2.2 Illustration of the excitation reference frame.

This $(\alpha,\beta)\rightarrow(d,q)$ transformation is expressed as,

$$\begin{bmatrix} X_{ds} \\ X_{qs} \end{bmatrix} = \begin{bmatrix} \cos(\theta_e) & \sin(\theta_e) \\ -\sin(\theta_e) & \cos(\theta_e) \end{bmatrix} \begin{bmatrix} X_{\alpha s} \\ X_{\beta s} \end{bmatrix}, \tag{2.8}$$

where the superscript denotes the new rotating reference frame. While the inverse $(d,q)\rightarrow(\alpha,\beta)$ transformation is given by,

$$\begin{bmatrix} X_{\alpha s} \\ X_{\beta s} \end{bmatrix} = \begin{bmatrix} \cos(\theta_e) & -\sin(\theta_e) \\ \sin(\theta_e) & \cos(\theta_e) \end{bmatrix} \begin{bmatrix} X_{ds} \\ X_{qs} \end{bmatrix}. \tag{2.9}$$

The magnetic equations in rotating frame are given by,

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$\lambda_{ds} = L_s i_{ds} + L_M i_{dr}, \tag{2.10}$$

$$\lambda_{qs} = L_s i_{qs} + L_M i_{qr}, \tag{2.11}$$

$$\lambda_{dr} = L_M i_{ds} + L_r i_{dr}, \tag{2.12}$$

$$\lambda_{qr} = L_M i_{qs} + L_r i_{qr}. \tag{2.13}$$

And the torque equation in (d,q) frame can be written as [3]

$$T_e = \frac{2n_p}{3R_r}\frac{L_m}{\tau_r}(i_{qs}\lambda_{dr} - i_{ds}\lambda_{qr}), \tag{2.14}$$

where τ_r is the rotor time constant

$$\tau_r = \frac{L_r}{R_r}, \tag{2.15}$$

and n_p is the number of pole pairs of the IM.

2.1.3 Rotor Flux Alignment

Since the (d,q) frame has been defined as rotating with the same angular velocity as the vector quantities of the motor, any one of these vectors can be used as a reference with which the (d,q) frame is to be aligned in order to further simplify the torque (2.14). In classical approach the (d,q) frame is aligned with rotor flux vector λ_r^s. This leads to

$$\lambda_{qr} = 0. \tag{2.16}$$

Then the torque equation becomes

$$T_e = \frac{2n_p}{3R_r}\frac{L_m}{\tau_r} i_{qs}\lambda_{dr} = K_T \lambda_{dr} i_{qs}, \tag{2.17}$$

where

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$K_T = \frac{2n_p}{3R_r}\frac{Lm}{\tau_r}. \tag{2.18}$$

Hence, when

$$\lambda_{qr} = Constant,$$

There is a linear relationship between current i_{qs} and torque.

The FOC schemes can be classified into two groups: the *direct field orientation* (DFO) and *indirect field orientation* (IFO). The DFO is based upon estimation of the rotor flux from the terminal voltage and current, while the IFO avoids the requirement of flux estimation by computing the appropriate motor slip frequency ω_{sl} to obtain the desired flux position θ_e:

$$\theta_e = \int(\omega_r + \omega_{sl})dt, \tag{2.19}$$

where

$$\omega_{sl} = \frac{L_m}{|\lambda_r|_{command}} * \frac{R_r}{L_r} * i_{qs}^{e*}. \tag{2.20}$$

Both the DFO and IFO, particularly IFO, are parameter sensitive. For example, inductance parameters vary about $\pm$20%, whereas rotor resistance changes dramatically ($\pm$100%) with temperature.

A typical closed loop IFO control scheme of IM is shown in Fig. 2.3. The objective of block "Controller" is to generate reference torque T_e^*. The block "i_{qs}^* Calculation" calculates the reference i_{qs}^* according to (2.17) as

$$i_{qs}^* = \frac{T_e^*}{K_T \lambda_{dr}^*} \tag{2.21}$$

where the reference rotor flux vector λ_{dr}^* is given by block "Phir*". The block "i_{dr}^* Calculation" calculates the reference i_{dr}^* by [3],

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

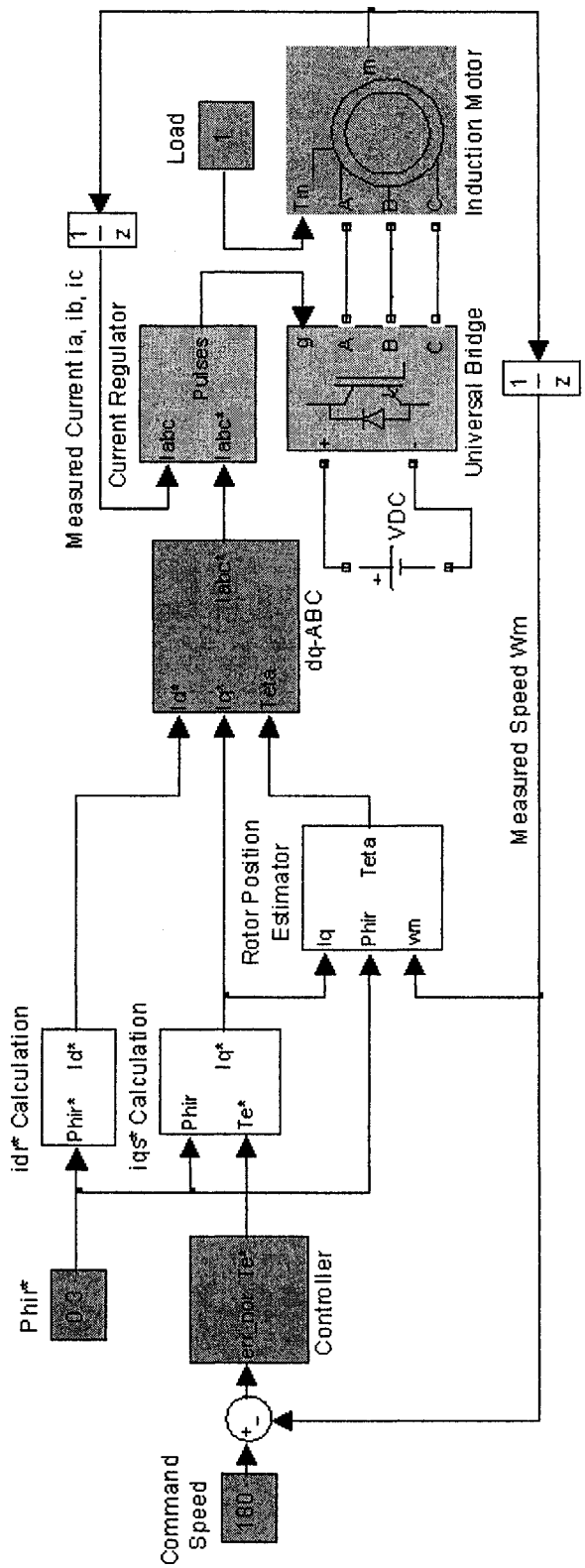


Fig. 2.3 A typical closed loop IFO control scheme of IM.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$i_{dr}^{*} = \frac{\lambda_{dr}^{*}}{Lm}. \tag{2.22}$$

The function of the block "Rotor Position Estimator" is to estimate the rotor flux position θ_e according to (2.19) and (2.20). The block "dq-ABC" does the inverse Park's and Clarke's transformation according to (2.9) and (2.2), and then outputs the reference 3-phase current. These reference 3-phase current is compared to the measured 3-phase current in the block "Current Regulator" which outputs switch signals for the inverter block "Universal Bridge".

2.2 Neuro-fuzzy Controller (NFC)

2.2.1 Basic Structure of Fuzzy Logic Controller

Fig. 2.4 shows the basic structure of a traditional fuzzy logic controller based system. Fuzzy logic controllers produce their command outputs from inputs based on the fuzzy rules. The fuzzy rule base contains a set of IF-THEN statements R. For a MISO (multi input single output) system,

$$R = \{R_1, R_2, ..., R_n\},$$

where the ith fuzzy logic rule is

$$R_i: \text{IF } x_1 \text{ is } A_1^{j1} \text{ AND } x_2 \text{ is } A_2^{j2} \text{ AND } \ldots \text{ AND } x_n \text{ is } A_n^{jn}, \text{ THEN } y \text{ is } M^j. \tag{2.23}$$

In expression (2.23) x_n is a crisp input variable, y is the command output variable, A_n^{jn} and M^j are fuzzy sets which affect the way how a human being describes the input x_n and output y. The input membership functions are formulas used to map the inputs to fuzzy sets with degree of membership in that set. So for a specific value $x_1(t)$ at time t,

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

it is mapped to the fuzzy set A_1^1 with degree $\mu_1^1(x_1(t))$ and to the fuzzy set A_1^2 with degree $\mu_1^2(x_1(t))$, and so on.

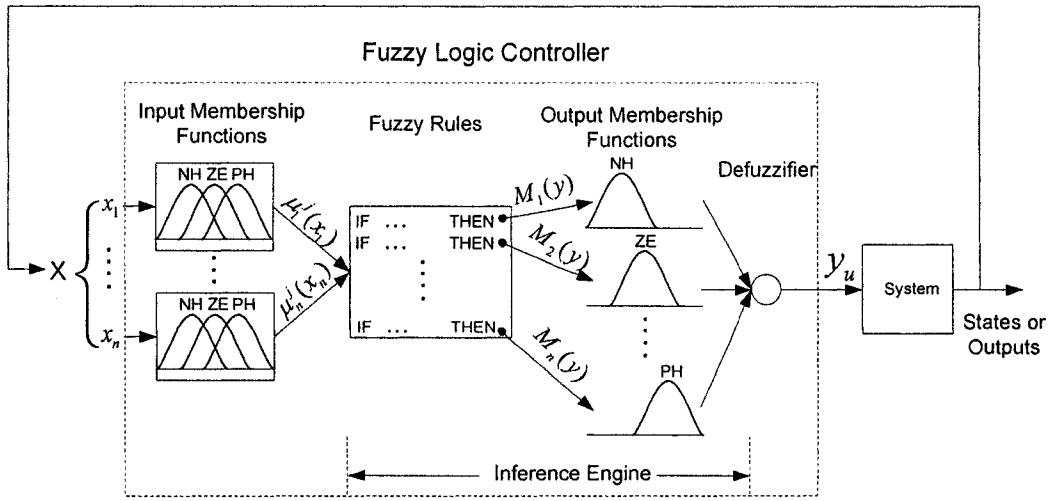


Fig. 2.4 The basic structure of a traditional fuzzy logic controller based system.

The inference engine in Fig 2.4 is to perform implication. For example, if there are two rules:

$$R_1: \text{IF } x_1 \text{ is } A_1^1 \text{ AND } x_2 \text{ is } A_2^1, \text{ THEN } y \text{ is } M^1,$$

$$R_2: \text{IF } x_1 \text{ is } A_1^2 \text{ AND } x_2 \text{ is } A_2^2, \text{ THEN } y \text{ is } M^2.$$

Then the firing strengths of rules R_1 and R_2 are defined as α_1 and α_2, respectively. Here α_1 is defined as

$$\alpha_1 = \mu_1^1(x_1) \wedge \mu_2^1(x_2), \tag{2.24}$$

where $\wedge$ is the *fuzzy AND* operation. The algebraic product is utilized as the *fuzzy AND* operation in the thesis. Hence, α_i, $i = 1,2$, becomes

$$\alpha_i = \mu_1^i(x_1) \wedge \mu_2^i(x_2) = \mu_1^i(x_1) \bullet \mu_2^i(x_2). \tag{2.25}$$

The above two rules, R_1 and R_2, lead to the corresponding decision with the membership function, $\hat{M}_i(y)$, $i = 1,2$, which is defined as

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$\hat{M}_i(y) = \alpha_i \wedge M_i(y) = \alpha_i \bullet M_i(y), \tag{2.26}$$

where y is the variable that represents the support values of the membership function. Combining the above decisions, we obtain the output decision

$$\hat{M}(y) = \hat{M}_1(y) \vee \hat{M}_2(y), \tag{2.27}$$

where $\vee$ is the is the *fuzzy OR* operation. The max (maximum) is utilized for *fuzzy OR* operation. Hence the output decision becomes

$$\hat{M}(y) = \hat{M}_1(y) \vee \hat{M}_2(y) = \max(\hat{M}_1(y), \hat{M}_2(y)). \tag{2.28}$$

Notice that the last result is the membership function curve. Before feeding the signal to the system, we need a defuzzification process to get a crisp decision, and the defuzzifier in Fig. 2.4 serves this purpose. Among the commonly used defuzzification strategies, the **centroid**, which returns the centre of area under the curve, yields a superior result. Let y_j be the support value at which the membership function, $\hat{M}_i(y)$, reaches the maximum value $\hat{M}_i(y)|_{y=y_j}$, then the defuzzification output is

$$y_u = \frac{\sum_j \hat{M}_j(y_j) y_j}{\sum_j \hat{M}_j(y_j)}. \tag{2.29}$$

2.2.2 Fuzzy Singleton Rule Based NFC

The section 2.2.1 describes the standard function operations, the so-called Mamdani's fuzzy inference method, in a traditional fuzzy logic control system. In spite of several well known advantages of fuzzy logic controller, the systematic procedure for the design of fuzzy logic controller is still lacking at present. The most straightforward approach is to define membership functions and rules subjectively by studying a human-operated system or an existing controller and then testing the

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

design for the proper output. The membership functions and/or rules should be adjusted if the design fails the test, or types of system are changed, or the control objectives are altered.

The progress made in Neural Network theory has made available self-learning procedures. One of the important characteristics of neural network is that neural network can adapt itself to changing control environment and can learn by the type of input and output. In fact as any fuzzy system can be viewed as a highly structured neural network, typical neural network learning techniques can be used to optimize the design of fuzzy logic controller. The combination of fuzzy logic with neural network leads to Neuro-Fuzzy controller.

However this combination brings heavy computational burden. The tuning of the parameters of membership functions and rules involves large amount of multi-plications and divisions. This fact has prevented the utilization of the neuro-fuzzy network into some industrial cost-sensitive field.

In this thesis, we adopt the Sugeno-Type Fuzzy Interference to speed up the computational process. The Sugeno-Type Fuzzy Interference, introduced in 1985, is similar to the Mamdani method in many respects. The first two parts of the fuzzy inference process, fuzzifying the inputs and applying the fuzzy operator, are exactly the same. The main difference between Mamdani and Sugeno is that the Sugeno output membership functions are either linear or constant.

A typical rule in a Sugeno fuzzy model has the form

$$R_1: \text{IF } x_1 \text{ is } A_1^1 \text{ AND } x_2 \text{ is } A_2^1, \text{ THEN } y = ax_1 + bx_2 + c.$$

For a zero-order Sugeno model, or *fuzzy singleton rules*, the output level y is a constant ($a = b = 0$). Such fuzzy singleton rules are in the following form:

$$R_1: \text{IF } x_1 \text{ is } A_1^1 \text{ AND } x_2 \text{ is } A_2^1, \text{ THEN } y = w_i.$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

The output level y of each rule is weighted by the firing strength α_i of the rule. For example, for an AND rule with x_1 is A_1^1 AND x_2 is A_2^1, the firing strength is

$$\alpha_1 = \mu_1^1(x_1) \bullet \mu_2^1(x_2), \tag{2.30}$$

where $\mu_1^1(.)$ and $\mu_2^1(.)$ are the membership functions for inputs x_1 and x_2. The final output of the system is the weighted average of all rule outputs, computed as

$$\text{Final Output} = \frac{\sum_{i=1}^{N} \alpha_i y_i}{\sum_{i=1}^{N} \alpha_i} \tag{2.31}$$

Because it is a more compact and computationally efficient representation than a Mamdani system, the Sugeno system lends itself to the use of adaptive techniques for constructing fuzzy models. These adaptive techniques can be used to customize the membership functions so that the fuzzy system best models the data. Also because of the linear dependence of each rule on the input variables, the Sugeno method is ideal for acting as an interpolating supervisor of multiple linear controllers that are to be applied, respectively, to different operating conditions of a dynamic nonlinear system.

Figure 2.6 shows network structure of the fuzzy singleton rules based NFC [74]. x_i is an input variable, y_u is the output variable, A_i^j are linguistic terms of the precondition part with membership functions $\mu_{A_i^j}(x_i)$, w_j is a real number of the rule part, $j = 1, 2, ..., M$ and $i = 1, 2, ..., n$.

Next we describe the functions of the nodes in each of the four layers. In the following, O_N^M is the out put of the Nth node in the Mth layer. From the network structure, it is note that the inputs of the O_N^M are coming from O_N^{M-1}. In the following equations, superscript is used to indicate the layer number.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

- ***Layer 1:*** The nodes in this layer just transmit input values to the next layer. That is,

$$O_i^I = x_i . \tag{2.32}$$

From the above equation, the link weight at layer one (w_i^1) is unity.

- ***Layer 2:*** The output function of the node in this layer is membership function. For example, for a isosceles triangular function as shown in Fig. 2.5

$$O_i^{II} = \mu_{A_i^j}(x_i) = M_{x_i}^j(c_i^j, b_i^j) = 1 - \frac{\left|O_i^I - c_i^j\right|}{b_i^j} \tag{2.33}$$

where c_i^j and b_i^j are, respectively, the center and the half of width of the isosceles triangular function of the jth term of the ith input linguistic variable x_i .

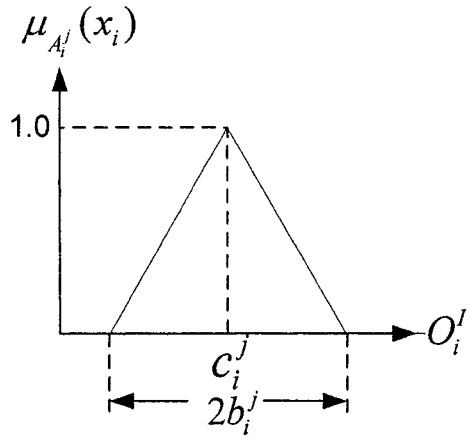


Fig. 2.5 The triangular function.

- ***Layer 3:*** The links in this layer are used to perform precondition matching of fuzzy logic rules. Hence the rule nodes should perform the fuzzy AND operation,

$$O_i^{III} = \prod_i^n O_i^{II} . \tag{2.34}$$

Then the link weight in layer three (w_i^3) is unity.

- ***Layer 4:*** The node in this layer performs defuzzification.

$$O^{IV} = \frac{\sum_{i=1}^{M} O_i^{III} w_i}{\sum_{i=1}^{M} O_i^{III}} . \tag{2.35}$$

From equation (2.32) - (2.35), the final output of the NFC can be found

$$y_u = \frac{\sum_{j=1}^{M} (\prod_{i=1}^{n} \mu_{A_i^j}(x_i)) w_j}{\sum_{j=1}^{M} (\prod_{i=1}^{n} \mu_{A_i^j}(x_i))} . \tag{2.36}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

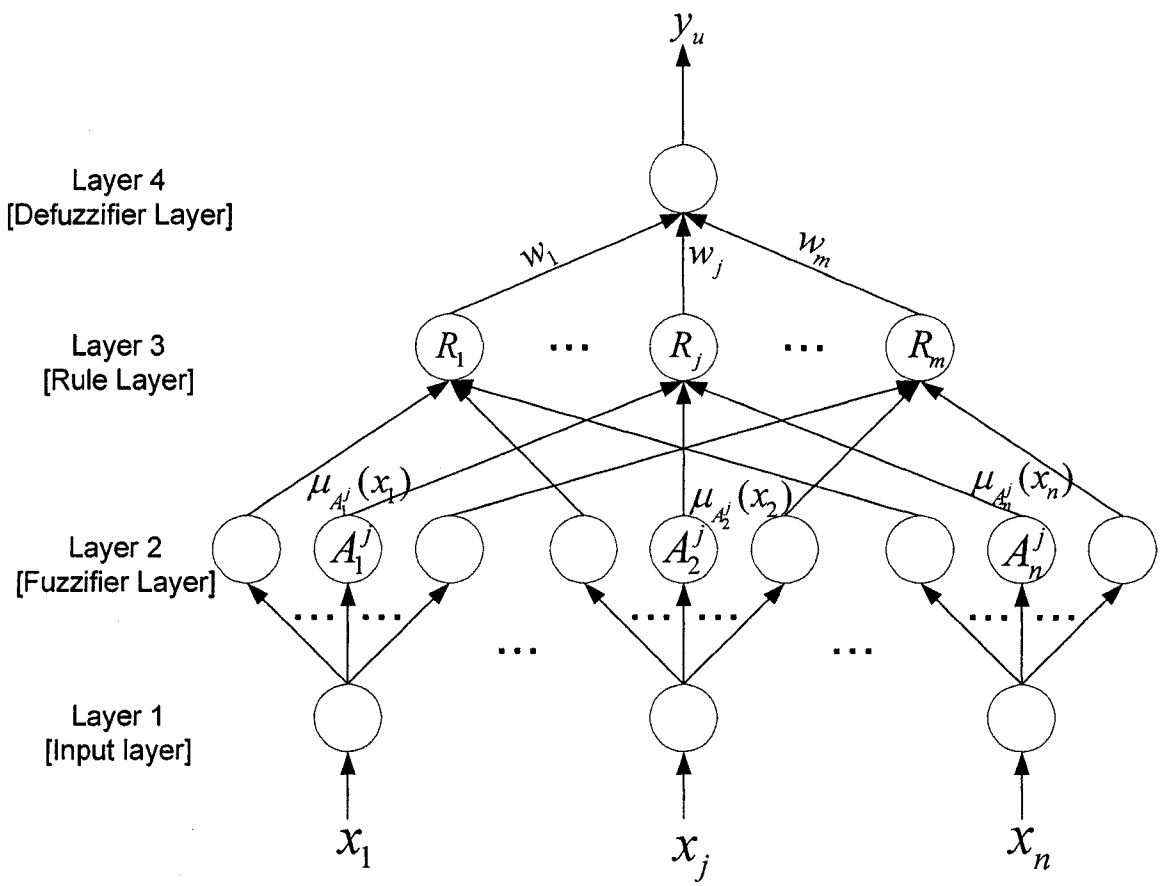


Fig. 2.6 Fuzzy singleton rules based NFC.

2.2.3 Parameters Tuning Methods

Parameter tuning of NFC with fuzzy singleton rules is the tuning of input membership function $\mu_{A_i^j}$ and the weights w_j.

2.2.3.1 Back Propagation (BP)

Back propagation was created to train the parameters of both nonlinear and linear, differentiable transfer functions and rules in a multiple-layer network. Standard backpropagation is a gradient descent algorithm. The training process requires a set of examples of proper network behavior - network inputs and corresponding target outputs. During training the weights (or rules in our fuzzy singleton rules based NFC)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

and parameters of membership functions of the network are iteratively adjusted to minimize the network performance function, which is usually defined as

$$E = \frac{1}{2}(y_u - y_u^d)^2, \tag{2.37}$$

where y_u^d is the desired output for input vector $X = (x_1, x_2, ..., x_n)^T$.

As an example, the tuning rules based on BP algorithm for the membership functions and weights can be written as,

$$c_i^j(n+1) = c_i^j(n) - \eta_c \frac{\partial E}{\partial c_i^j}, \tag{2.38}$$

$$b_i^j(n+1) = b_i^j(n) - \eta_b \frac{\partial E}{\partial b_i^j}, \tag{2.39}$$

$$w_j(n+1) = w_j(n) - \eta_w \frac{\partial E}{\partial w_j}. \tag{2.40}$$

The derivatives can be found by the chain rule

$$\frac{\partial E}{\partial c_i^j} = \frac{\partial E}{\partial y_u} \frac{\partial y_u}{\partial \mu_{A_i^j}} \frac{\partial \mu_{A_i^j}}{\partial c_i^j}, \tag{2.41}$$

$$\frac{\partial E}{\partial b_i^j} = \frac{\partial E}{\partial y_u} \frac{\partial y_u}{\partial \mu_{A_i^j}} \frac{\partial \mu_{A_i^j}}{\partial b_i^j}, \tag{2.42}$$

$$\frac{\partial E}{\partial w_j} = \frac{\partial E}{\partial y_u} \frac{\partial y_u}{\partial w_j}. \tag{2.43}$$

The hypothesis behind BP algorithm is that properly trained BP networks tend to give reasonable answers when presented with inputs that they have never seen. Typically, a new input leads to an output similar to the correct output for input vectors used in training that are similar to the new input being presented. This generalization property makes it possible to train a network on a representative set of input/target

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

pairs and get good results without training the network on all possible input/output pairs.

2.2.3.2 Recursive Least-Squares (RLS)

From equation(2.36), it is observed that for fuzzy singleton rules, the final inferred output y_u is a linear function of the consequent parameters w_j. Hence, given the values of the membership parameters and p training data ($X^{(k)}, y_u^{d(k)}$), $k = 1,2,...,p$, we can form p linear equations in terms of the consequent parameters as follows:

$$\begin{bmatrix} y_u^{d(1)} \\ y_u^{d(2)} \\ \cdot \\ \cdot \\ \cdot \\ y_u^{d(p)} \end{bmatrix} = \begin{bmatrix} \bar{u}_1^{(1)} w_1 + \bar{u}_2^{(1)} w_2 + ... + \bar{u}_M^{(1)} w_M \\ \bar{u}_1^{(2)} w_1 + \bar{u}_2^{(2)} w_2 + ... + \bar{u}_M^{(2)} w_M \\ \cdot \\ \cdot \\ \cdot \\ \bar{u}_1^{(p)} w_1 + \bar{u}_2^{(p)} w_2 + ... + \bar{u}_M^{(p)} w_M \end{bmatrix} = \begin{bmatrix} \bar{u}_1^{(1)} & \bar{u}_2^{(1)} & ... & \bar{u}_M^{(1)} \\ \bar{u}_1^{(2)} & \bar{u}_2^{(2)} & ... & \bar{u}_M^{(2)} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \bar{u}_1^{(p)} & \bar{u}_2^{(p)} & ... & \bar{u}_M^{(p)} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \cdot \\ \cdot \\ \cdot \\ w_M \end{bmatrix}, \quad (2.44)$$

Where the $\bar{\mu}_j^{(k)} = \dfrac{\prod_{i=1}^{n} \mu_{A_i^j}(x_i)}{\sum_{j=1}^{M} (\prod_{i=1}^{n} \mu_{A_i^j}(x_i))}$ and $\prod_{i=1}^{n} \mu_{A_i^j}(x_i)$ value are calculated, for example, from equation (2.33) when the input is $X^{(k)}$.

Introducing vectors

$$\varphi^T(i) = \begin{bmatrix} \bar{u}_1^{(i)} & \bar{u}_2^{(i)} & ... & \bar{u}_M^{(i)} \end{bmatrix}, \quad (2.45)$$

$$\Phi = \begin{bmatrix} \varphi^T(1) & \varphi^T(2) & ... & \varphi^T(p) \end{bmatrix}^T, \quad (2.46)$$

$$\theta = \begin{bmatrix} w_1 & w_2 & ... & w_M \end{bmatrix}^T, \quad (2.47)$$

$$Y = \begin{bmatrix} y_u^{d(1)} & y_u^{d(2)} & ... & y_u^{d(p)} \end{bmatrix}^T, \quad (2.48)$$

equation (2.44) can be expressed in a matrix-vector form

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$Y = \Phi\theta. \tag{2.49}$$

According to linear equation (2.49) the tuning of w_j can be viewed as a problem of parameters estimation. The least squares method is a useful technique for parameter estimation. And this method is particularly simple if the model has the property of being linear in the parameters such as (2.49). If $\Phi^T\Phi$ is non-singular, the parameter should be

$$\hat{\theta} = (\Phi^T\Phi)^{-1}\Phi^T Y \tag{2.50}$$

So as to minimize the least-squares loss function

$$V(\theta, p) = \frac{1}{2}\sum_{i=1}^{p}(y_u^{d(i)} - \varphi^T(i)\theta). \tag{2.51}$$

In many cases, the row vectors of matrix Φ and the corresponding elements in Y are obtained sequentially in real time. Therefore it is desirable to compute the least squares estimation of θ recursively by using the following formula:

$$\hat{\theta}(n) = \hat{\theta}(n-1) + K(n)(y_u^{d(n)} - \varphi^T(n)\hat{\theta}(n-1)), \tag{2.52}$$

$$K(n) = P(n)\varphi(n) = P(n-1)\varphi(n)(I + \varphi^T(n)P(n-1)\varphi(n))^{-1}, \tag{2.53}$$

$$\begin{aligned} P(n) &= P(n-1) - P(n-1)\varphi(n)(I + \varphi^T(n)P(n-1)\varphi(n))^{-1}\varphi^T(n)P(n-1) \\ &= (I - K(n)\varphi^T(n))P(n-1) \end{aligned}. \tag{2.54}$$

2.2.3.3 Kaczmarz's Projection Algorithm

The RLS algorithm given by equation (2.52) - (2.54) has two sets of state variables, $\hat{\theta}$ and P, which must be updated at each step. For large n the updating of the matrix P dominates the computing effort. There are several simplified algorithms that avoid updating the P matrix at the cost of slower convergence. Kaczmarz's *projection algorithm* [91] is one simple solution. It gives

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$\hat{\theta}(n) = \hat{\theta}(n-1) + \frac{\gamma \varphi^T(n)}{\varphi^T(n)\varphi(n)} (y_u^{d(n)} - \varphi^T(n)\hat{\theta}(n-1)), \qquad (2.55)$$

where $0 < \gamma < 2$ is a factor used to change the step length of the parameter adjustment.

2.2.4 NFC in Control Systems

A major property of NFC for control systems is their ability to generate input-output maps that can approximate any function with desired accuracy. The performance of a NFC in control systems is to large extent dependent on how the NFC is trained. However an important issue related to the training of NFC has raised much concern. This issue is that the desired output of the network (the appropriate controller-generated control input to the plant) is not readily available but has to be induced from the known desired output. Some solutions are

- Inverse dynamics of the plant;
- Differentiating a model;
- Reinforcement learning [56], [74].

2.2.4.1 Inverse Dynamics of the Plant

Fig. 2.7 (a) shows how a neural network can be used to identify the inverse of a plant, where the input to the network is the output of the plant and the target output of the network is the plant input. Once one has such an inverse, it can be used for control purpose as shown in Fig. 2.7 (b). The desired plant output is provided as input to the network, and the resulting network output is then used as input to the plant. Since the network is a plant inverse, this plant input causes the desired plant

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

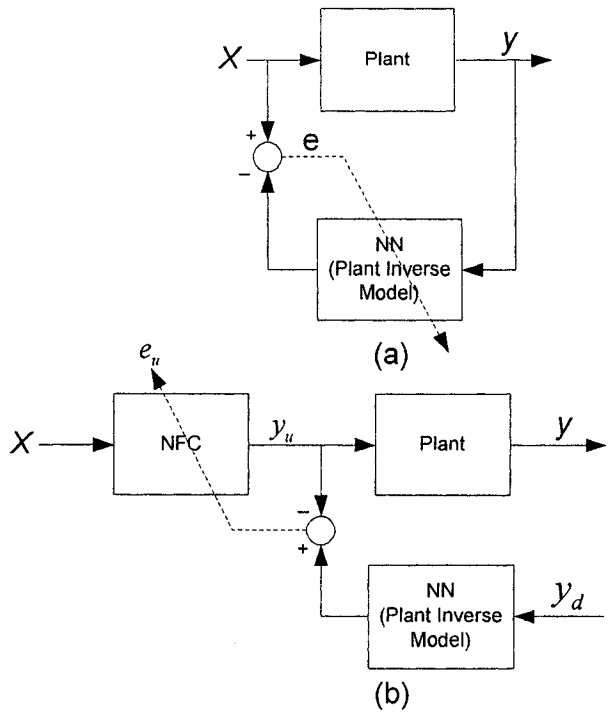


Fig. 2.7 (a) Training of the plant inverse model, (b) Application of the plant inverse model.

output. A major problem with inverse identification is that the plant's inverse is not always well-defined. That is, more than one plant input produces the same output. In this case, the network attempts to map the same network input to many different target responses. As a result, it tends to average over the various targets, thereby producing a mapping that is not necessarily an inverse.

2.2.4.2 Differentiating a Model

Fig. 2.8 shows the basic control scheme. Since the error at the output of the NFC (controller error e_u) is not directly available and only the system error e_y can be measured at the output of plant, the objective function to be minimized by the NFC is defined as,

$$E = \frac{1}{2}(y_d - y)^2 . \tag{2.56}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Thus the equation (2.41) - (2.43) become

$$\frac{\partial E}{\partial c_i^j} = -\frac{\partial E}{\partial y}\frac{\partial y}{\partial y_u}\frac{\partial y_u}{\partial \mu_{A_i^j}}\frac{\partial \mu_{A_i^j}}{\partial c_i^j}, \tag{2.57}$$

$$\frac{\partial E}{\partial b_i^j} = -\frac{\partial E}{\partial y}\frac{\partial y}{\partial y_u}\frac{\partial y_u}{\partial \mu_{A_i^j}}\frac{\partial \mu_{A_i^j}}{\partial b_i^j}, \tag{2.58}$$

$$\frac{\partial E}{\partial w_j} = -\frac{\partial E}{\partial y}\frac{\partial y}{\partial y_u}\frac{\partial y_u}{\partial w_j}. \tag{2.59}$$

The term $\frac{\partial y}{\partial y_u}$, known as Jacobian matrix, corresponds to the forward gain of the plant. To find the derivatives of the equation (2.57) - (2.59) we need to know the Jacobian matrix of the plant. This usually implies that we need a model for the plant

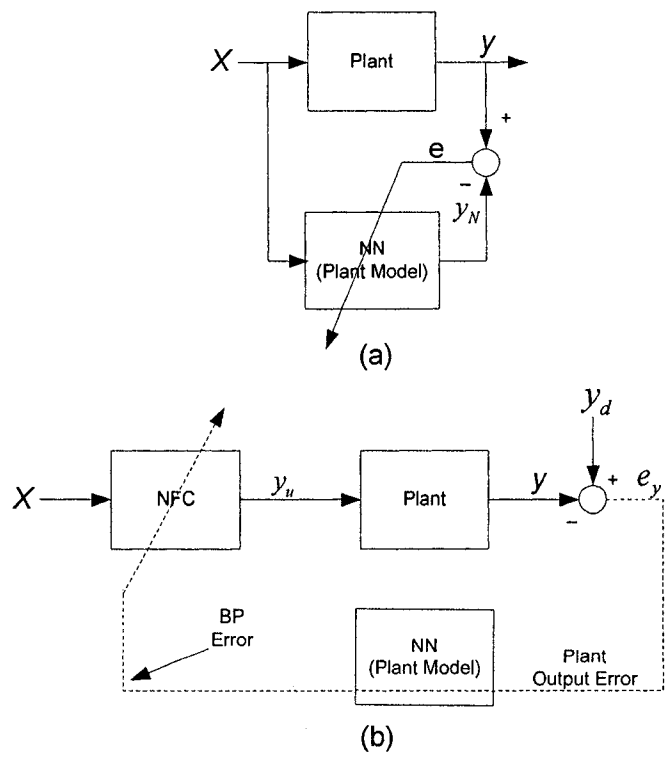


Fig. 2.8 (a) Training of the plant model, (b) Application of the plant model.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

and the Jacobian matrix obtained from the model, which could be a neural network [42], neural fuzzy network (NFN) [59], or another appropriate mathematical description of the plant [49], [50].

As shown in Fig. 2.8 (a) a multilayer network is first trained to identify the plant's forward model, then this network and another NFN used as the controller are configured as in Fig. 2.8 (b). The advantage of a forward model having this form is that one can efficiently compute the derivative of the model's output with respect to its input by means of the BP process, which evaluates the transpose of the network Jacobian matrix at the network's current input vector. As a result, propagating errors between actual and desired plant outputs back through the forward model produces error in the control signal, which can be used to train another NFN to be a controller. This error back-propagation path is illustrated by the dash line in Fig. 2.8 (b). This method has advantages over the direct identification of a plant inverse when the inverse is not well-defined.

2.2.4.3 Reinforcement Learning

The reinforcement learning is proposed to overcome the difficulty encountered by supervised learning. The supervised learning schemes require precise training data to indicate the exact desired output, and then use the precise training data to compute the output errors for training the whole network. Unfortunately, such detailed and precise training data may be very expensive or even impossible to obtain in some real word applications because the controlled systems may only be able to provide the learning algorithm with an "evaluative" reinforcement feedback such as a binary direction of right/wrong of the current controller output.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

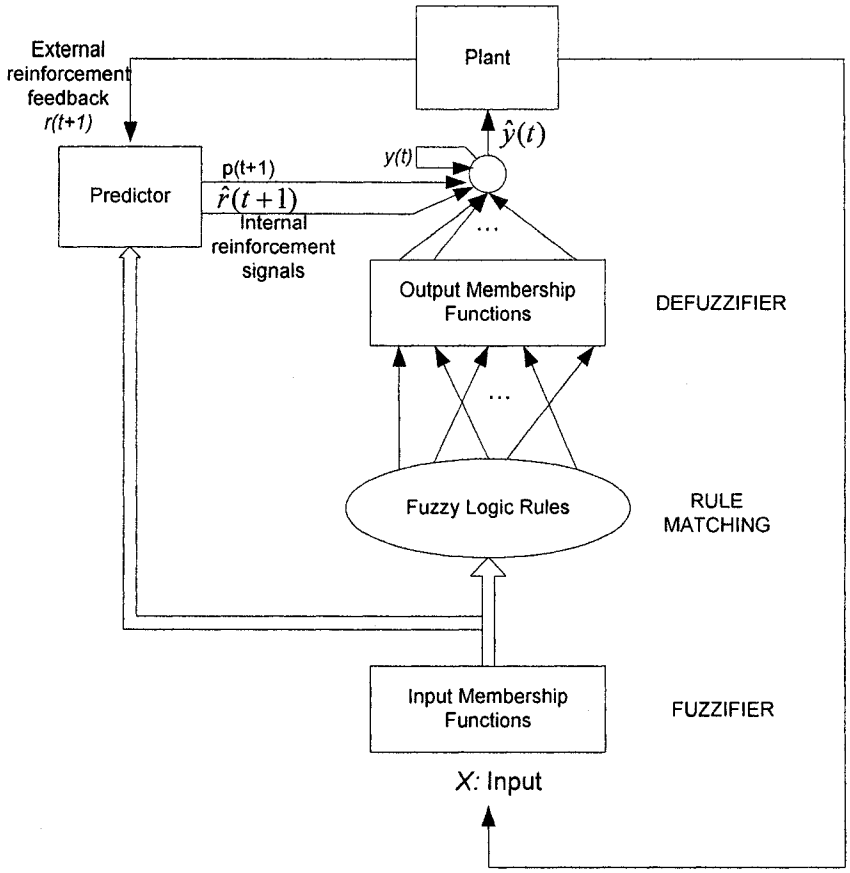


Fig. 2.9 The reinforcement control scheme.

The reinforcement control scheme is more complicated than the conventional NFC as illustrated in Fig. 2.9 [74]. The *Predictor* is a NN or NFN function as an internal evaluator capable of evaluating the plant performance. Another NFN functions as an action network. These two networks share the same input vector and membership functions. The action network output y does not directly act on the plant. Instead, it is treated as mean (expected) action. The actual action, $\hat{y}$, is chosen by exploring a range around this mean point. This range of exploring corresponds to the variance of a probability function which is the normal distribution in our design. This amount of exploration, $\sigma(t)$, can be chosen as

$$\sigma(t) = \frac{k}{1+e^{2p(t)}}, \tag{2.60}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

where k is a search-range scaling constant which can be simply set to 1, and $p(t)$ is the predicted (expected) reinforcement signal used to predict $r(t)$. Equation (2.60) is a monotonic decreasing function between k and 0, and $\sigma(t)$ can be interpreted as the extent to which the output node searches for a better action. Since $p(t)$ is the expected reward signal, if $p(t)$ is small, the exploratory range, $\sigma(t)$, will be large according to (2.60). On the contrary, if $p(t)$ is large, $\sigma(t)$ will be small. This amounts to narrowing the search about the mean, $y(t)$, if the expected reinforcement signal is large. This can provide a higher probability to choose an actual action, $\hat{y}$, which is very close to $y(t)$, since it is expected that the mean action $y(t)$ is very close to the best action possible for the current given input vector. On the other hand, the search range about the mean $y(t)$ is broadened if the expected reinforcement signal is small such that the actual action can have a higher probability of being quite different from the mean action $y(t)$. Once the variance has been decided, the actual output can be set as:

$$\hat{y}(t) = K * N(y(t), \sigma(t)). \tag{2.61}$$

K is a scaling factor to fit the specifications of the controlled plant. That is, $\hat{y}(t)$ is a normal or Gaussian random variable with the density function:

$$f(\hat{y}) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\hat{y}-y)^2}{2\sigma^2}}. \tag{2.62}$$

The gradient information is estimated as [74]:

$$\frac{\partial r}{\partial y} \approx [r(t) - p(t)]\left[\frac{\hat{y}(t-1) - y(t-1)}{\sigma(t-1)}\right]. \tag{2.63}$$

In(2.63), the term $\frac{\hat{y}-y}{\sigma}$ is the normalized difference between the actual and expected actions, $r(t)$ is the real reinforcement feedback for the actual action $\hat{y}(t-1)$, and $p(t)$ is the predicted reinforcement signal the expected action $y(t-1)$. Equation (2.63) was

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

derived based on the following intuitive concept. If $r(t) > p(t)$, then $\hat{y}(t-1)$ is a better action than the expected one, *y(t-1)* and *y(t-1)* should be moved closer to $\hat{y}(t-1)$. If $r(t) < p(t)$, then $\hat{y}(t-1)$ is a worse action than the expected one, and *y(t-1)* should be moved farther away from $\hat{y}(t-1)$. This idea also comes from the observation of a discrete gradient descent method. The concept behind (2.63) is frequently adopted in the stochastic exploration technique.

After the gradient information is available, the reinforcement learning has been transformed to the supervised learning.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Chapter 3

Development & Implementation of a Low Computational Neuro-Fuzzy Speed Controller for IM Drive

In this chapter a novel and low computational self-tuned NFC is developed for speed control of an IM drive. The proposed NFC combines fuzzy logic and a four-layer artificial neural network (ANN) scheme. The proposed NFC lower down the computational burden by using only speed error as the input instead of using both speed error and its derivative which are widely employed as inputs by the conventional NFCs. The simple structure of the proposed NFC makes it easier to be implemented in practical applications. Based on the knowledge of vector control and Back Propagation (BP) algorithm a supervised self-tuning method is developed to adjust membership functions and weights of the proposed NFC. The complete drive incorporating the proposed self tuned NFC is experimentally implemented using a

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

digital signal processor board DS-1104 for a laboratory 1/3 hp motor. The effectiveness of the proposed NFC based IM drive is tested both in simulation and experiment at different operating conditions. Performance comparisons between the developed NFC, conventional NFC and PI controller are also provided. Comparison of results in simulation and experiment proves that the simplification of the proposed NFC does not decrease the system performance.

3.1 Control Structure

The schematic diagram of the proposed NFC-based indirect field oriented control of induction motor is shown in Fig. 3.1. The basic configuration of the drive system consists of an induction motor fed by a current controlled voltage source inverter. The normalized speed error $\Delta\omega\%$ is processed by the NFC to generate the reference torque $T_e^*(n)$. The command current $i_{QS}^{e*}(n)$ is calculated from equation (2.17) as following,

$$i_{Qs}^{e*}(n) = T_e^*(n)\frac{3R_r}{2n_p}\frac{\tau_r}{L_m}\frac{1}{\lambda_{Dr}^{e*}}. \tag{3.1}$$

Currents i_{QS}^{e*} and i_{DS}^{e*} are transformed into i_a^*, i_b^* and i_c^* according to inverse Clarke's and Park's transformation as,

$$\begin{bmatrix} i_a^* \\ i_b^* \\ i_c^* \end{bmatrix}_{(n)} = \begin{bmatrix} \frac{2}{3} & 0 \\ -\frac{1}{3} & \frac{1}{\sqrt{3}} \\ -\frac{1}{3} & -\frac{1}{3} \end{bmatrix} \begin{bmatrix} \cos(\theta_e) & -\sin(\theta_e) \\ \sin(\theta_e) & \cos(\theta_e) \end{bmatrix}_{(n)} \begin{bmatrix} i_{Ds}^{e*} \\ i_{Qs}^{e*} \end{bmatrix}_{(n)}. \tag{3.2}$$

The $\theta_e(n)$ is calculated indirectly based on (1.20) and (1.21) as,

$$\theta_e(n) = \theta_e(n-1) + [\omega_r(n) + \omega_{sl}(n)]T_s, \tag{3.3}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

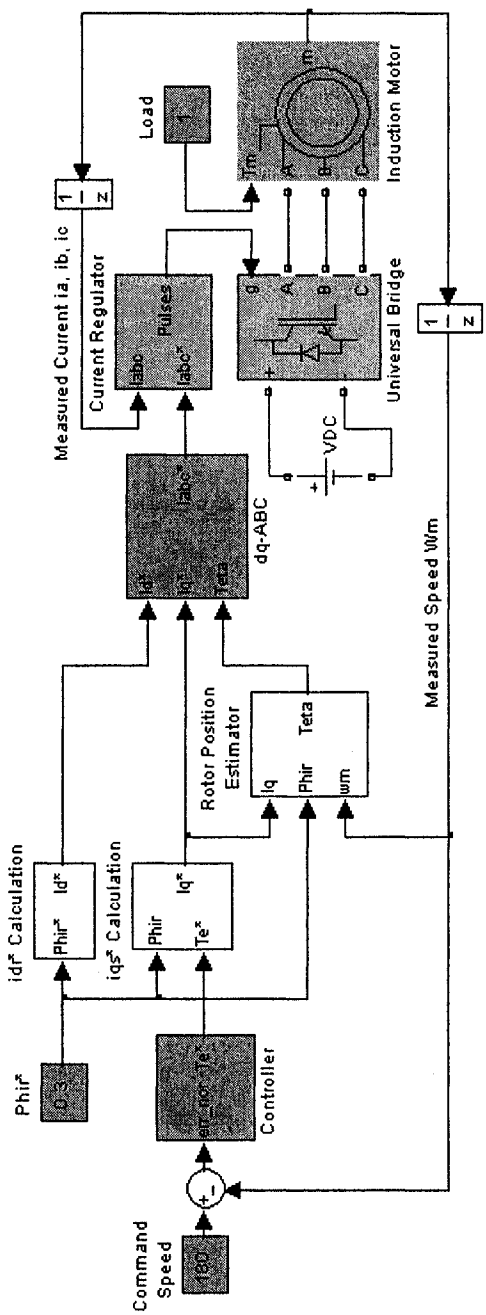


Fig. 3.1 Block diagram of the proposed NFC based IM drive.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

where T_s is sampling time and

$$\omega_{sl}(n) = \frac{L_m}{|\lambda_r|_{command}} * \frac{R_r}{L_r} * i_{Qs}^{e*}(n). \tag{3.4}$$

The phase command currents i_a^*, i_b^* and i_c^* are then compared with the corresponding actual currents i_a, i_b and i_c. Current error signals Δi_a, Δi_b and Δi_c are then applied to the hysteretic current controllers to generate PWM logic signals, which are used to fire the power semiconductor switches of the 3-phase inverter. Thus the inverter produces the actual voltages to run the induction motor. This PWM control technique is called *Current-Controlled Voltage Source Inverters* which is illustrated in Fig. 3.2. The input-output characteristics of the phase-A hysteretic current controller is shown in Fig. 3.3. The width of the hysteresis loop, denoted by *h*, represents the tolerance bandwidth for the controlled current. If the current error, Δi_a, is greater than *h/2*, i.e., current i_a is unacceptably lower than the reference current, i_a^*, the within the tolerance band. The other two controllers operate in a similar manner. Thus the switches of the 3-phase inverter are controlled on the following logic,

if $\Delta i_a > \frac{h}{2}$, then Sa = 1, T1 is ON, i_a increase;

if $\Delta i_a < -\frac{h}{2}$, then Sa′= 1, T4 is ON, i_a decrease;

if $\Delta i_b > \frac{h}{2}$, then Sb =1, T3 is ON, i_b increase;

if $\Delta i_b < -\frac{h}{2}$, then Sb′=1, T6 is ON, i_b decrease;

if $\Delta i_c > \frac{h}{2}$, then Sc = 1, T5 is ON, i_c increase;

if $\Delta i_c < -\frac{h}{2}$, then Sc′= 1, T2 is ON, i_c decrease.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

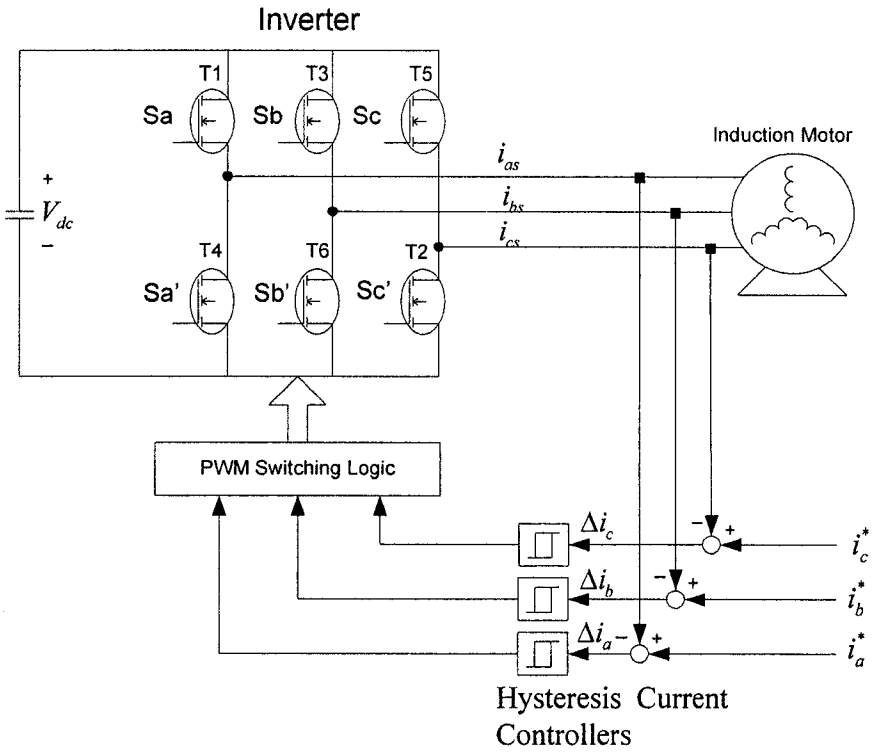


Fig. 3.2 Block diagram of current-controlled VSI.

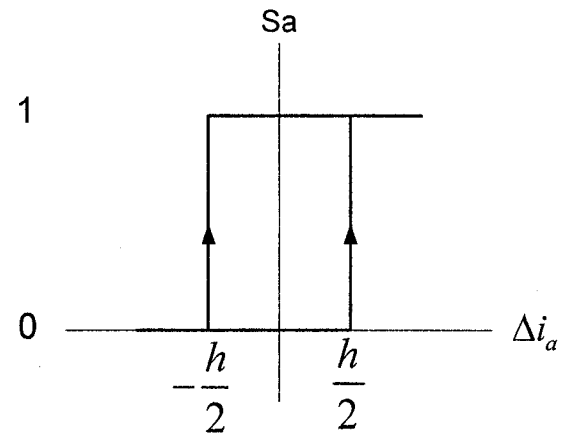


Fig. 3.3 The input-output characteristics of hysteretic current controller.

The width, h, of the tolerance band affects the switching frequency of the inverter. The narrower is the band, the higher frequency in the switching and hence

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

the better quality of the current. However, due to the limitation of the switching frequency of the semiconductor devices, the hysteretic band is chosen as 0.05 for real-time experiment in this thesis.

3.2 Design of the Neuro-Fuzzy Controller

The proposed NFC incorporates fuzzy logic and a learning algorithm with a four-layer artificial neural network (ANN) structure as depicted in Fig. 3.4. The learning algorithm modifies the NFC to closely match the desired system performance. The detailed discussions on different layers of the NFC are given below.

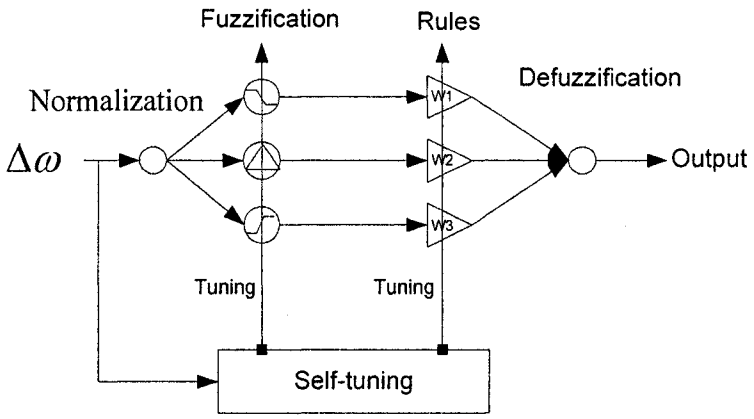


Fig. 3.4 Structure of the NFC.

Input Layer:

The input of the proposed NFC is the normalized speed error, which is given as,

$$O^I = \frac{\omega^* - \omega}{\omega^*} * 100\%, \tag{3.5}$$

where ω is the measured speed, ω^* is the command speed, I denotes the 1st layer.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Fuzzification Layer:

A three membership function-based fuzzy set is utilized to obtain the fuzzy number for the input. In order to keep the computational burden low, in the proposed NFC, triangular and trapezoidal functions are chosen as the membership functions as shown in Fig. 3.5.

The node equations in this layer are given as,

$$O_1^{II} = \begin{cases} 1 & x_i^{II} \le b_1 \\ \dfrac{x_i^{II} - a_1}{b_1 - a_1} & b_1 < x_i^{II} < a_1 \\ 0 & x_i^{II} \ge a_1 \end{cases}, \tag{3.6}$$

$$O_2^{II} = \begin{cases} 0 & \left|x_i^{II}\right| \ge b_2 \\ 1 - \dfrac{x_i^{II} - a_2}{b_2} & \left|x_i^{II}\right| < b_2 \end{cases}, \tag{3.7}$$

$$O_3^{II} = \begin{cases} 0 & x_i^{II} \le a_3 \\ \dfrac{x_i^{II} - a_3}{b_3 - a_3} & a_3 < x_i^{II} < b_3 \\ 1 & x_i^{II} \ge b_3 \end{cases}. \tag{3.8}$$

where x_i^{II} is the input of the 2nd layer which is same as the output of the 1st layer. It is considered that a_2=0 in order to further lower the computational burden.

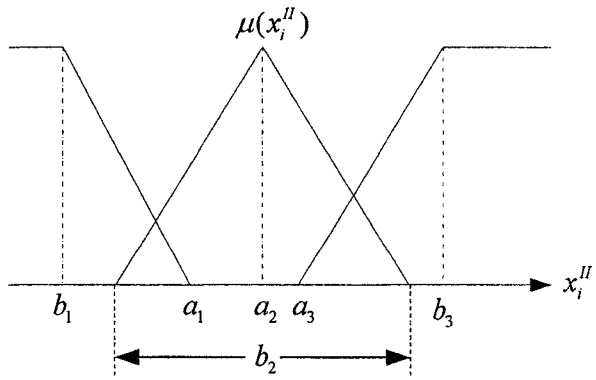


Fig. 3.5 Membership functions for input.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Rule Layer:

No "AND" logic is needed in the rule layer since there is only one input in the input layer. The node equations in rule layer are specified as,

$$O_i^{III} = x_i^{III} w_j = O_i^{II} w_j, \tag{3.9}$$

Where x_i^{III} is the input of the 3rd layer which is same as the output of 2nd layer.

Defuzzification Layer:

The center of gravity method is used to determine the output of NFC. The node equation is specified as,

$$y = O_i^{VI} = \frac{\sum x_i^{VI}}{\sum O_i^{II}} = \frac{\sum O_i^{III}}{\sum O_i^{II}} = \frac{\sum O_i^{II} w_i}{\sum O_i^{II}}, \tag{3.10}$$

where x_i^{VI} is the input of the 4th layer which is same as the output of 3rd layer.

In order to compare with the conventional 2-input NFC, we adopt the conventional 2-input NFC structure in [49] as shown in Fig. 3.6 and implement it in simulation and experiment. The NFC in [49] employed normalized speed error and its derivative as two inputs and has been successfully used for high performance control of IM. This NFC has 9 rules and 9 weights and hence the computational overhead is high.

3.3 On-Line Self-Tuning Algorithm

Since it is impossible to determine or calculate the desired NFC's output i_{QS}^{e*} and find train data off-line covering all operating conditions, a kind of supervised on-line self-tuning method is introduced in this thesis. Instead of using desired controller's output i_{QS}^{e*} as target, the system error is used to assess the performance of

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

controller

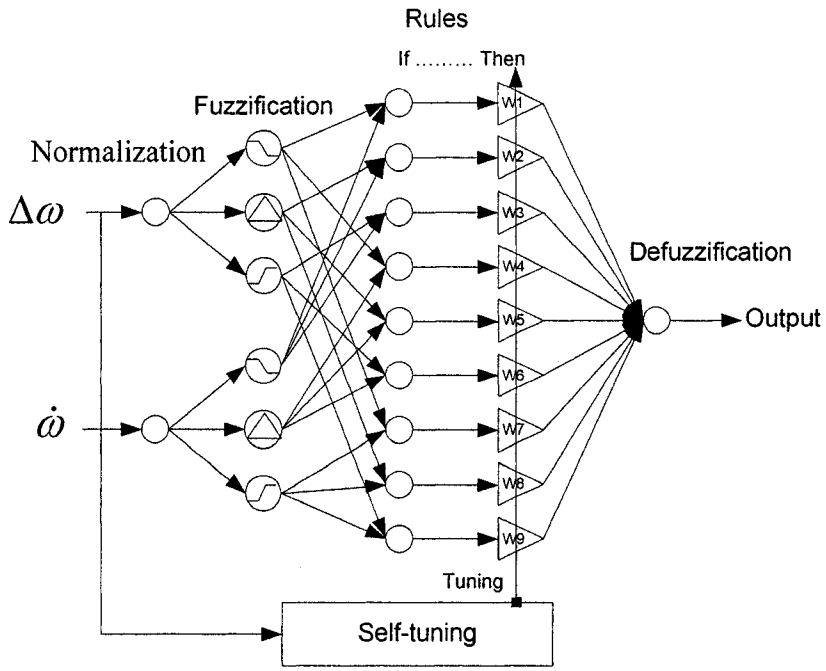


Fig. 3.6 Conventional NFC.

and evaluate the current state of system, and then to guide the control action in the right direction.

The objective function to be minimized is defined by

$$E = \frac{1}{2}r^2 = \frac{1}{2}\left(\omega^* - \omega\right)^2. \tag{3.11}$$

Hence, the learning rules can be derived as follows [74],

$$a_i(n+1) = a_i(n) - \eta_{a_i}\frac{\partial E}{\partial a_i}, \tag{3.12}$$

$$b_i(n+1) = b_i(n) - \eta_{b_i}\frac{\partial E}{\partial b_i}, \tag{3.13}$$

$$w_j(n+1) = w_j(n) - \eta_{w_j}\frac{\partial E}{\partial w_j}, \tag{3.14}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

where $\eta_{a_i}, \eta_{b_i}, \eta_{w_j}$ are the learning rates of the corresponding parameters. The derivatives can be found by chain rule as,

$$\frac{\partial E}{\partial a_i} = \frac{\partial E}{\partial r}\frac{\partial r}{\partial \omega}\frac{\partial \omega}{\partial y}\frac{\partial y}{\partial O_i^{II}}\frac{\partial O_i^{II}}{\partial a_i}, \tag{3.15}$$

$$\frac{\partial E}{\partial b_i} = \frac{\partial E}{\partial r}\frac{\partial r}{\partial \omega}\frac{\partial \omega}{\partial y}\frac{\partial y}{\partial O_i^{II}}\frac{\partial O_i^{II}}{\partial b_i}, \tag{3.16}$$

$$\frac{\partial E}{\partial w_j} = \frac{\partial E}{\partial r}\frac{\partial r}{\partial \omega}\frac{\partial \omega}{\partial y}\frac{\partial y}{\partial w_j}, \tag{3.17}$$

where the common parts of equations (20)-(22) are as follows,

$$\frac{\partial E}{\partial r} = r = \omega^* - \omega, \tag{3.18}$$

$$\frac{\partial r}{\partial \omega} = -1, \tag{3.19}$$

$$\frac{\partial \omega}{\partial y} = J(t), \tag{3.20}$$

where $J(t)$ is the Jacobean matrix of the system [74]. The Jacobean matrix *J(t)* is not easily found directly. Thanks to the FOC the IM system can be viewed as a single input single output system, then the *J(t)* can be estimated as a constant value $K_J > 0$ [76], [77].

From equations (3.6) - (3.20), the update rules can be determined as follows [75],

$$w_j(n) = w_j(n-1) + \eta_{w_j} K_J r(n) \frac{O_i^{II}(n-1)}{\sum O_j^{II}}, \tag{3.21}$$

$$a_1(n+1) = a_1(n) - \eta_{a_1} K_J r(n) \frac{w_1(n)}{\sum O_j^{II}} \frac{1 - O_1^{II}(n)}{b_1(n) - a_1(n)}, \tag{3.22}$$

$$b_1(n+1) = b_1(n) - \eta_{b_1} K_J r(n) \frac{w_1(n)}{\sum O_j^{II}} \frac{O_1^{II}(n)}{b_1(n) - a_1(n)}, \tag{3.23}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$b_2(n+1) = b_2(n) + \eta_{b2} K_J r(n) \frac{w_2(n)}{\sum O_j^{II}} \frac{1 - O_2^{II}(n)}{b_2(n)}, \tag{3.24}$$

$$a_3(n+1) = a_3(n) - \eta_{a_3} K_J r(n) \frac{w_3(n)}{\sum O_j^{II}} \frac{1 - O_3^{II}(n)}{b_3(n) - a_3(n)}, \tag{3.25}$$

$$b_3(n+1) = b_3(n) - \eta_{b_3} K_J r(n) \frac{w_3(n)}{\sum O_j^{II}} \frac{O_3^{II}(n)}{b_3(n) - a_3(n)}. \tag{3.26}$$

In our control scheme, we set $\eta_{a_1} = \eta_{a_3} = \eta_{b_1} = \eta_{b_2} = \eta_{b_3}$.

Based on these update rules, the following steps are employed for tuning the parameters of a_1, a_3, b_1, b_2, b_3 and w_j [74]:

Step 1: First an initial set of fuzzy logic rules and initial values of a_1, a_3, b_1, b_2, b_3 and w_j are selected.

Step 2: The normalized speed error is calculated, which is input to the NFC.

Step 3: Fuzzy reasoning is performed for the input data. The membership values O_i^{II} are then calculated by using (3.6)-(3.8).

Step 4: Tuning of the weights w_j of the consequent part is performed by using (3.21).

Step 5: Tuning of the a_1, a_3, b_1, b_2 and b_3 is done by substituting the tuned real number w_j obtained in step 4, the measured reinforcement signal r, the membership value O_i^{II} into equations (3.22) - (3.26).

Step 6: Repeat from step 3.

3.4 Simulation results

Before implementing in real time, the performance of the proposed NFC based IM drive is investigated extensively in simulation. A simulation model of the proposed NFC is developed in Matlab/Simulink [72] software as shown in Fig. 3.7.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

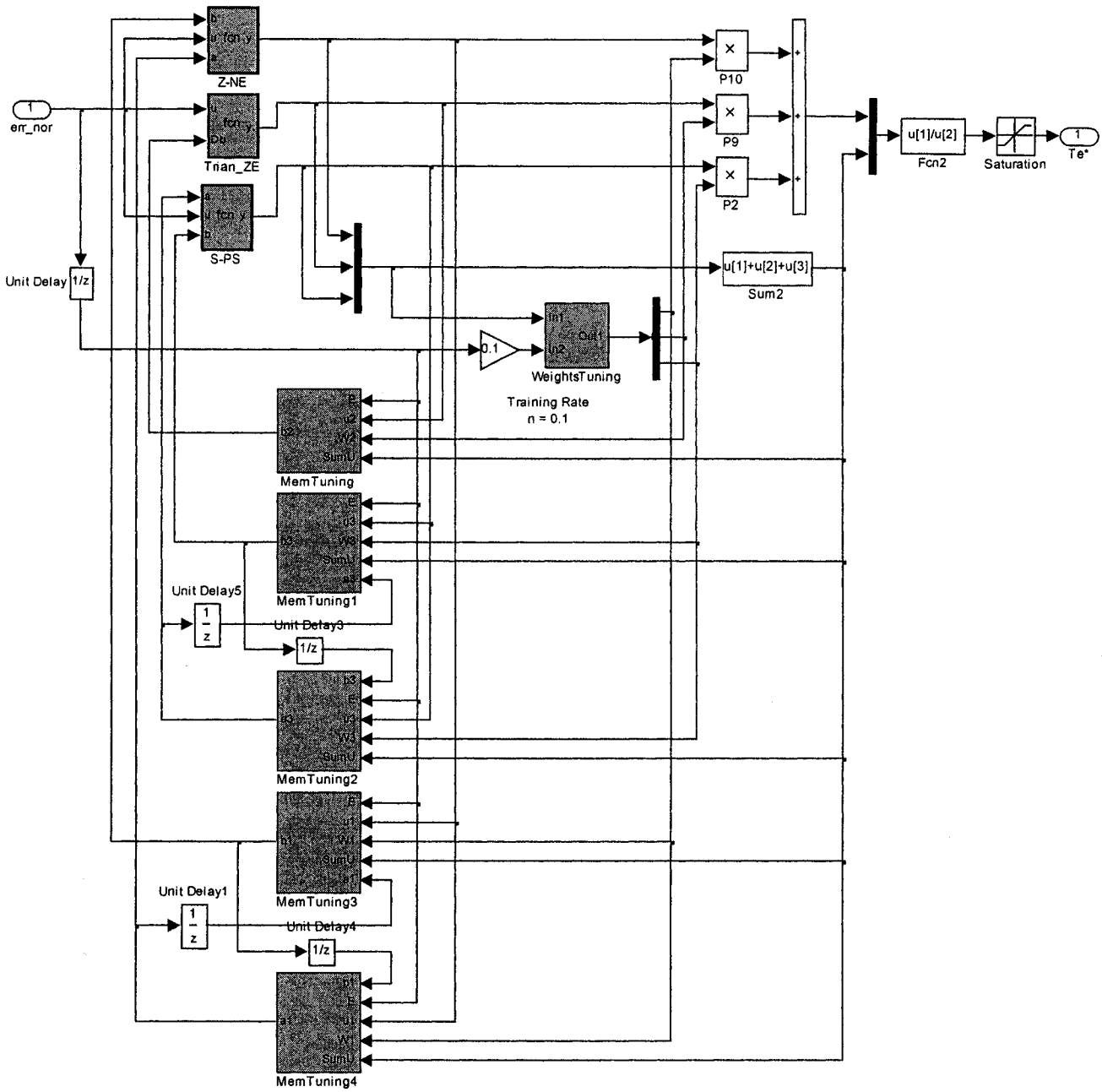


Fig. 3.7 Simulation model of the proposed NFC.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

The performance of the proposed NFC is compared to a conventional 2-input NFC [49] and a tuned PI controller. The proportional gain K_p is set as 0.1, and the integral gain K_i is set as 0.35 after trial and error so that the overall performance of PI controller is comparable with both of the NFCs. The parameters of the IM utilized for simulations are listed in Appendix A-1.

Fig. 3.8 (a)-(c) show the simulated starting speed response of the drive at rated load and rated speed for the proposed NFC, conventional NFC and PI controller, respectively. As seen from Fig. 3.8 (c), both of the NFCs show zero overshoot and less settling time than PI controller. Fig. 3.9(a)-(c) show the corresponding stator currents of the drive. Fig. 3.9(c) shows that for PI controller the starting current is higher as compared to both of NFCs. Fig. 3.10(a)-(c) show the corresponding torque responses of the drive. Obviously, the PI controller needs more torque and hence more current to start the motor.

Fig. 3.11 show the speed responses of the drive system with a step increase in load from zero to rated level using the proposed NFC, conventional NFC and PI controllers, respectively. In this test, the proposed NFC and conventional NFC show less dip in speed and less settling time than PI controller. The proposed NFC has a longer settling time than the conventional NFC, but smaller dip in speed.

Fig. 3.12 (a)-(c) show the speed responses of the drive system first with a step decrease on the command speed from 180rad/sec to 150rad/sec, and then a step increase on the command speed from 150rad/sec to 180rad/sec using the proposed NFC, conventional NFC and PI controllers, respectively. In this test, the proposed NFC has a little larger undershoot than PI controller, but no overshoot and less settling time.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Fig. 3.13 (a)-(c) show the simulated starting speed response of the drive for the proposed NFC, conventional NFC and PI controller, respectively, when the rotor resistance has been risen to be two times of the rated one.

Fig. 3.14 (a)-(c) show the corresponding torque response. Compared with Fig. 3.8 and Fig. 3.9, it is noted that the transient torque response of PI controller is displaying oscillation which leads to higher overshoot and longer settling time in the speed response. While the proposed NFC shows fairly same performances. This proves the robust ability of the proposed NFC.

Based upon tests, it is evident that the proposed NFC has overall better performance in terms of speed overshoot, dropdown and tracking over conventional PI controller. It also shows that the proposed NFC does not decrease system performance significantly as compared to the conventional 2-input and 9-membership functions NFC. These simulation results prove that the proposed NFC has no trade off between simplification and performance decreasing.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

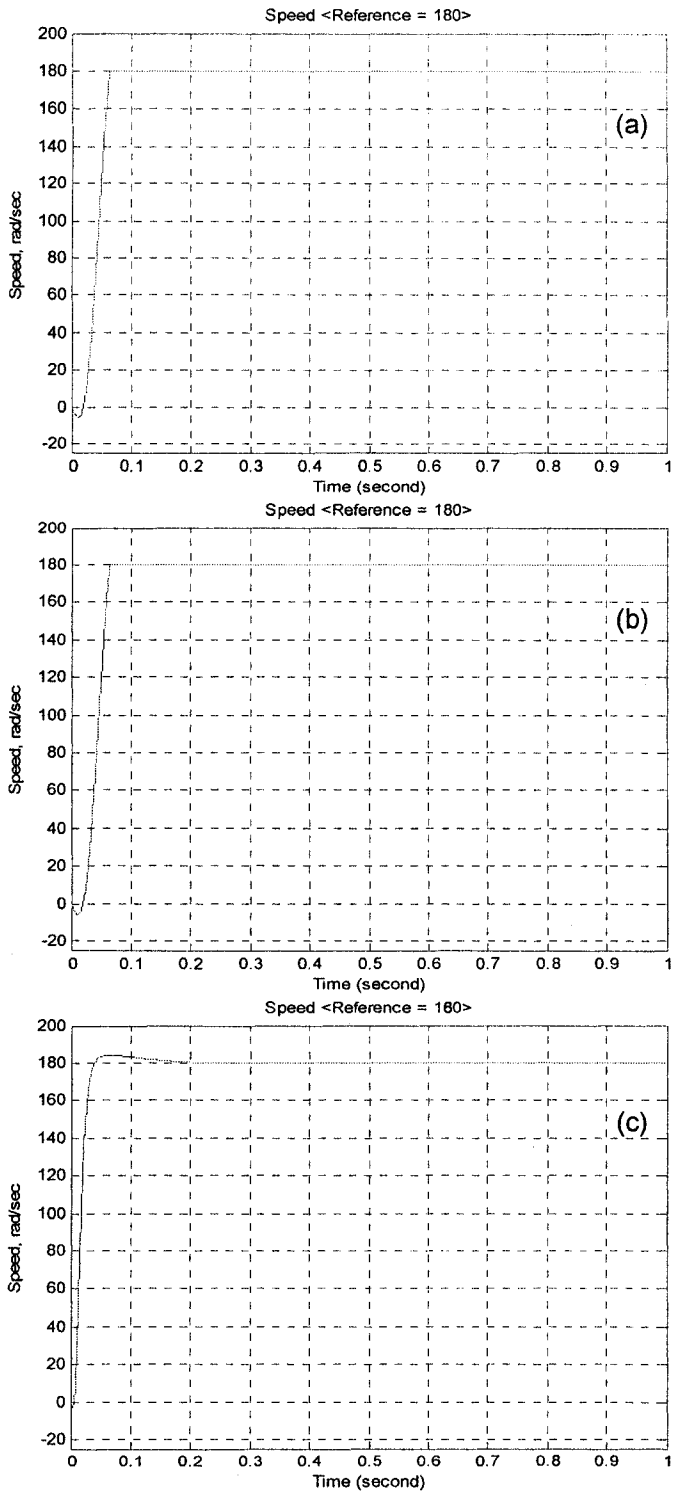


Fig. 3.8 Simulated starting speed responses of the drive: (a) Proposed NFC, (b) 2-input NFC, (c) PI.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

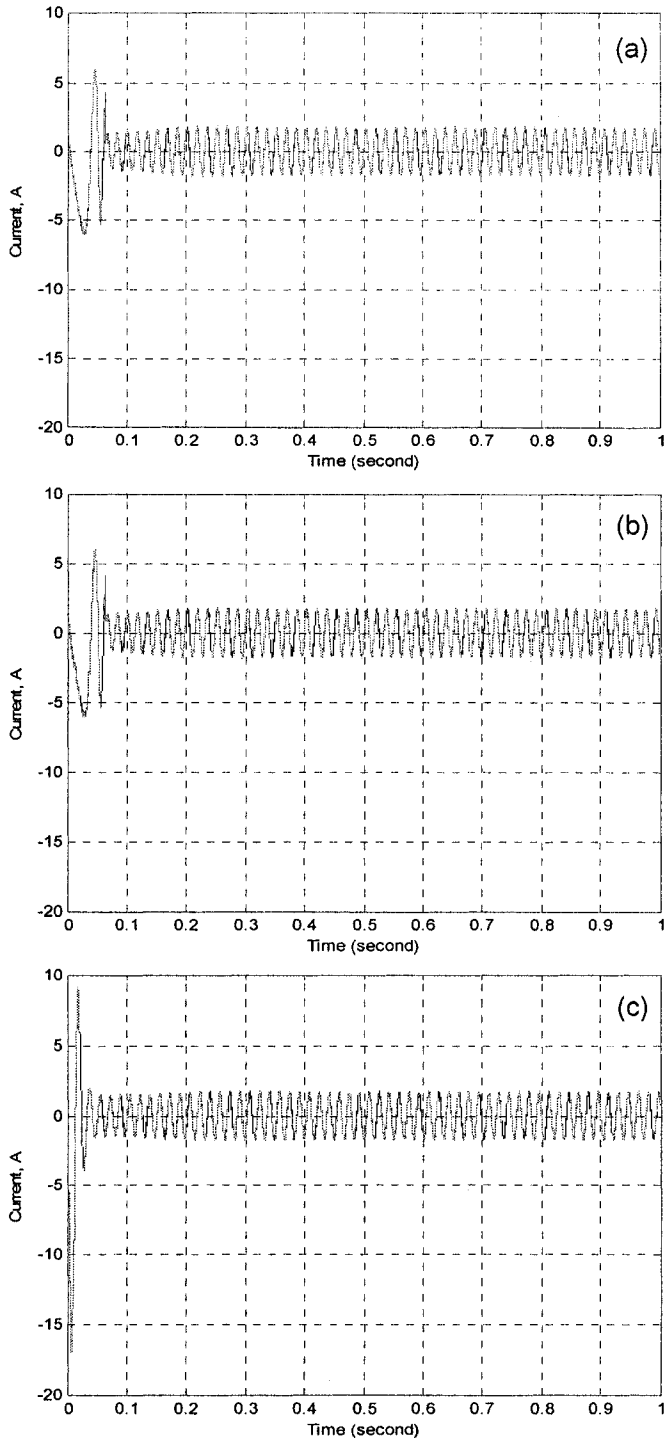


Fig. 3.9 Simulated starting current of the drive: (a) proposed NFC, (b) 2-input NFC, and (c) PI.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

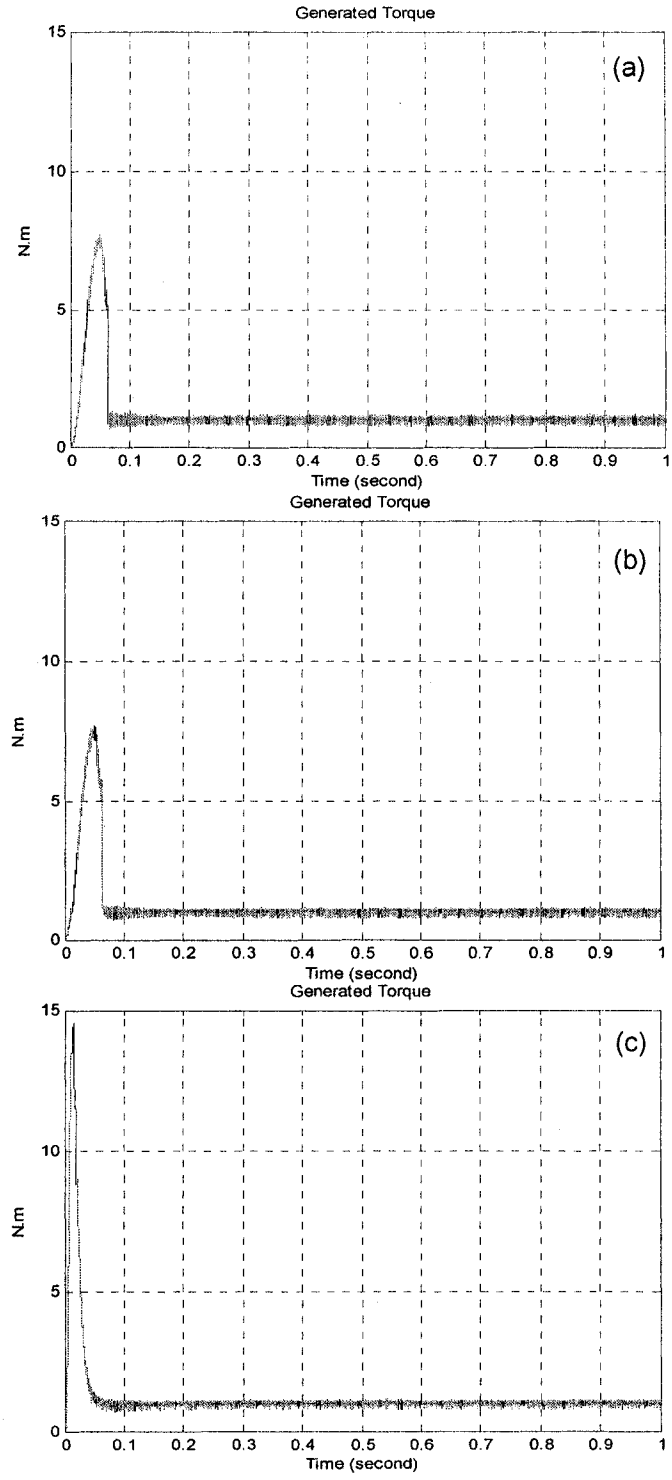


Fig. 3.10 Simulated starting Torque of the drive: (a) proposed NFC, (b) 2-input NFC, (c) PI.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

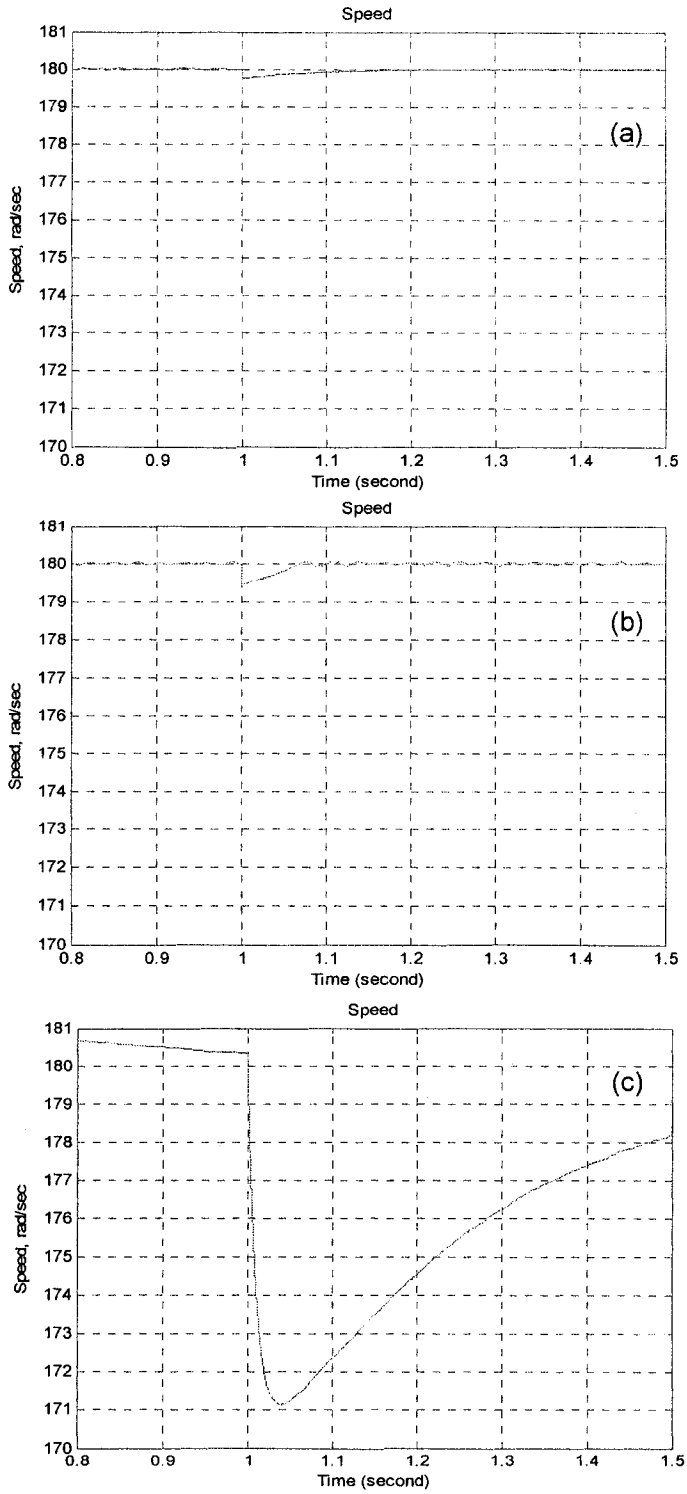


Fig. 3.11 Simulated speed responses at a step change of load: (a) proposed NFC, (b) conventional 2-input NFC, (c) PI.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

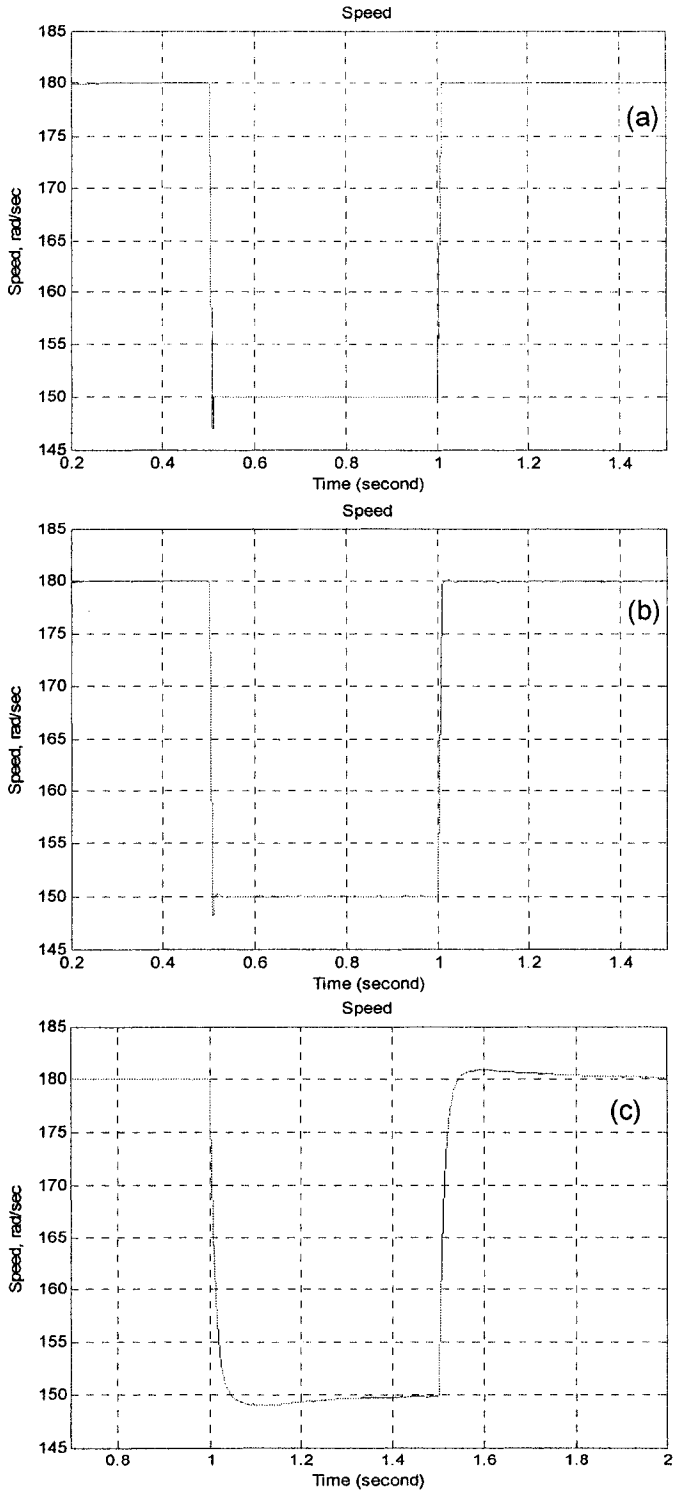


Fig. 3.12 Simulated speed responses at a step change of speed reference: (a) proposed NFC, (b) conventional 2-input NFC, (c) PI.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

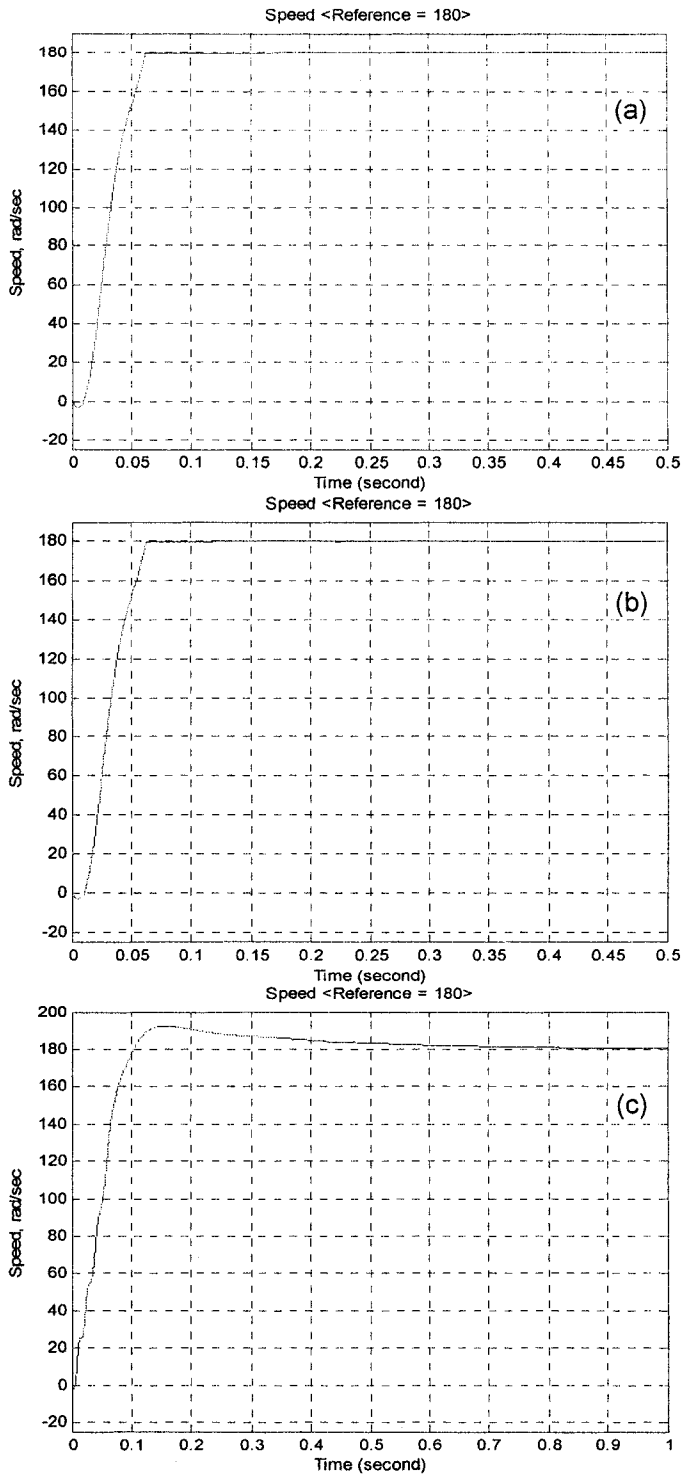


Fig. 3.13 Simulated starting speed responses of the drive due to rotor resistance variation: (a) proposed NFC, (b) 2-input NFC, (c) PI.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

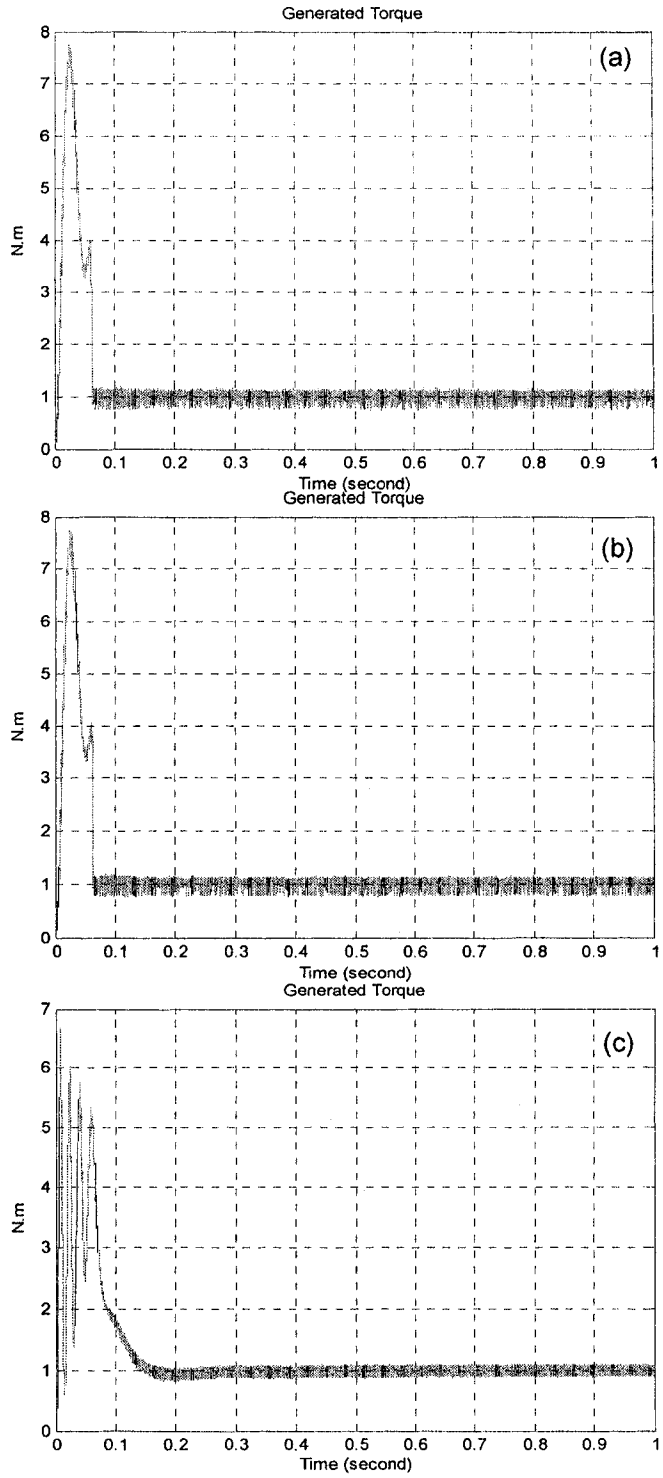


Fig. 3.14 Simulated starting Torque responses of the drive due to rotor resistance variation: (a) proposed NFC, (b) 2-input NFC, (c) PI.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

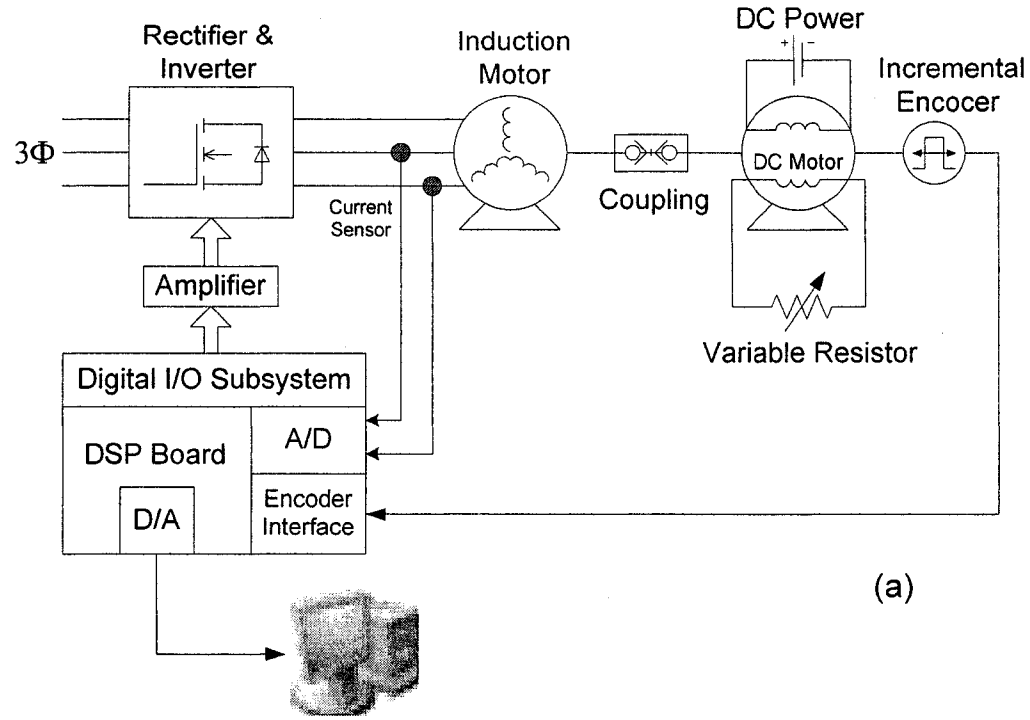
3.5 Experimental Implementations & Results

3.5.1 Drive Set-up

The proposed self tuned NFC based vector control of IM drive system has been implemented in real-time using the DSP board DS1104 [73]. This board is mainly based on 64-bit floating-point MPC8240 processor with PPC603e core. The block diagram and the photograph of the experimental system are shown in Fig. 3.15 (a) and (b), respectively. The PC-based controller produces numerical switching commands sent to DSP board and the outputs of the DSP board are sent to the amplifier circuit to drive the VSI inverter. The actual motor currents are measured by the Hall-effect sensors and fed back to the DSP board through the A/D channels. The Rotor position is sensed by an optical incremental encoder of 1000-line resolution and is fed back to the DSP board through the encoder interface. The test IM is coupled to a dc machine. The dc machine is operated as a generator in order to adjust load to the IM.

The NFC and self-tuning algorithm are implemented through developing a real-time Simulink model as shown in Appendix D. Then the model is compiled and downloaded to the DSP board utilizing ControlDesk software and real-time workshop (RTW). Since the proposed NFC has a simple structure, the highest sampling frequency can reach up to *14.3 kHz*. For comparison purpose, a PI-controller-based system is also developed and experimentally implemented. In real-time, the proportional gain, K_p, and the integral gain, K_i, are readjusted by trial and error to be 0.1 and 0.02, respectively, so that there is no steady-state error and the settling time, overshoot, undershoot can be comparable to the proposed NFC. For the NFC, the tuning rate for weights is chosen to be 0.1, and the tuning rate for membership

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.



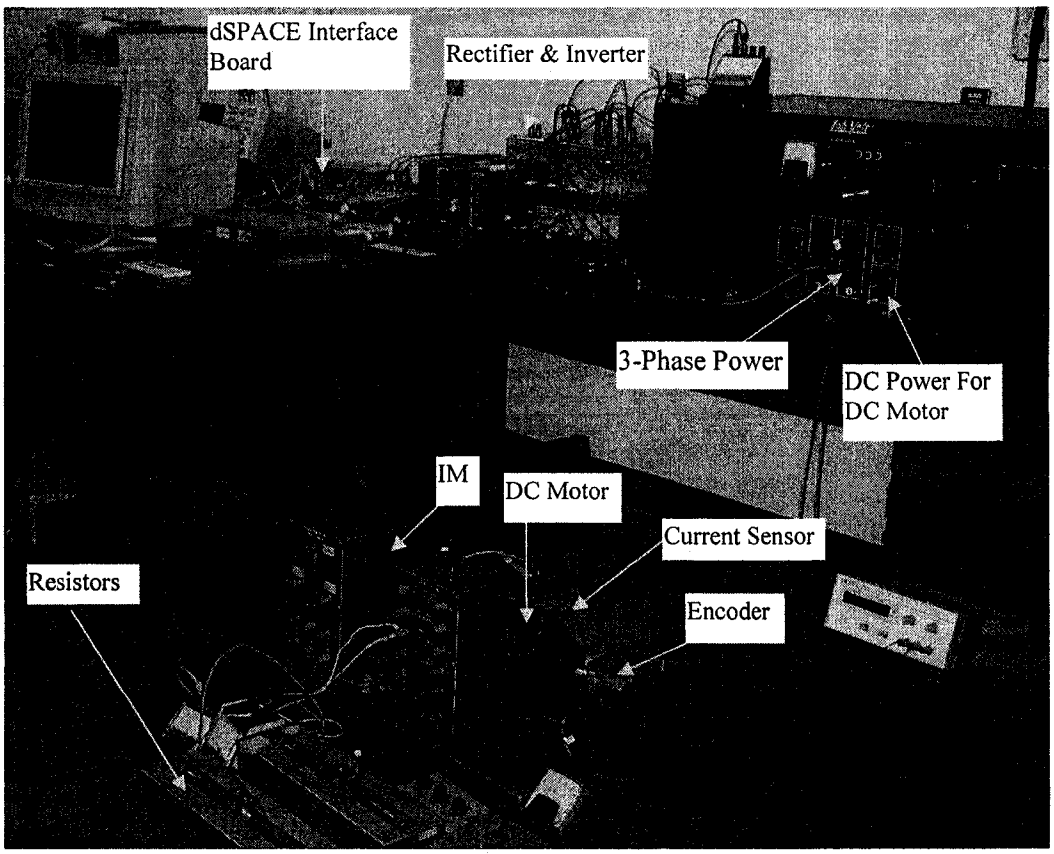


Fig. 3.15 (a) Block diagram of the experimental setup, (b) Photograph of the laboratory experimental setup.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

functions is chosen to be 0.008. The small values of tuning rates are chosen so that there will be a smooth transition from one state to another. A conventional 2-input NFC based system is also developed and experimentally implemented. To make a fair comparison, we change the membership functions in [49] are changed in similar way as proposed NFC, but with fixed parameters. Same modification is done to the self-tuning method. For fair comparison, the sampling frequency utilized for both of the NFCs and PI controller was 10 k*Hz*.

3.5.2 Experimental Results

The simulated results are verified by the experimental results. The experimental data is sampled by DSP board and filtered to eliminate the speed ripple which is caused by unbalance of power source and / or IM structure. Sample experimental results are presented below.

Fig. 3.16 (a)-(c) show the no load experimental starting speed response of the drive at command speed of 150rad/s for the proposed NFC, conventional NFC and PI controller, respectively. As seen from Fig. 3.16 (a)-(c), the PI controller shows bigger overshoot and longer settling time compared to NFCs. Both of the NFCs show the almost same performance.

Fig. 3.17 (a)-(c) show the no load experimental speed responses of the drive system first with a step decrease on the command speed from 150rad/sec to 120rad/sec, and then a step increase on the command speed from 120rad/sec to 150rad/sec for the proposed NFC, conventional NFC and PI controllers, respectively. In this test, both of the NFCs show the almost same performance.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

show the almost same performance, no overshoot and short settling time. Whereas the PI controller shows bigger overshoot and longer settling time compared to NFCs.

Fig. 3.19 (a)-(c) show the experimental speed responses of the drive system first with a step decrease on the command speed from 150rad/sec to 120rad/sec, and then a step increase on the command speed from 120rad/sec to 150rad/sec with an approximate 20% rated level load for the proposed NFC, conventional NFC and PI controllers, respectively. In this test, both of the NFCs show the almost same performance. During command speed step increasing, both the NFCs show invisible overshoot, but PI controller shows a bigger overshoot.

Fig. 3.20 (a)-(c) illustrate the test on the step increase/decrease of load from zero to mostly 20% of the rated level for the proposed NFC, conventional NFC and PI controllers, respectively. In this test, both of the NFCs show much less overshoot/undershoot in speed as compared to PI controller.

Figures 3.22 (a)-(b) illustrate the comparison of the reference current and measured current during the load increase/decrease. It can be seen that the actual current is following the reference current well.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

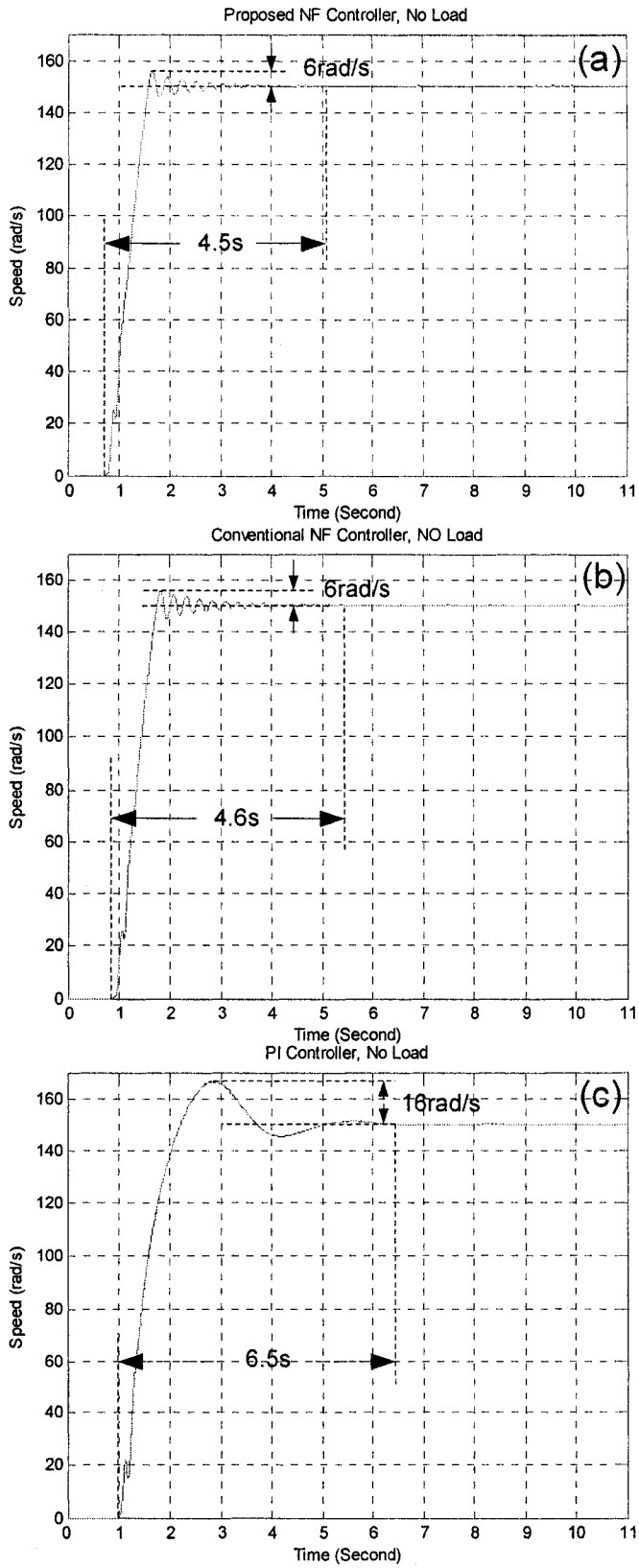


Fig. 3.16 Experimental starting speed responses of the drive at no load: (a) proposed NFC, (c) 2-input NFC, (C) PI.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

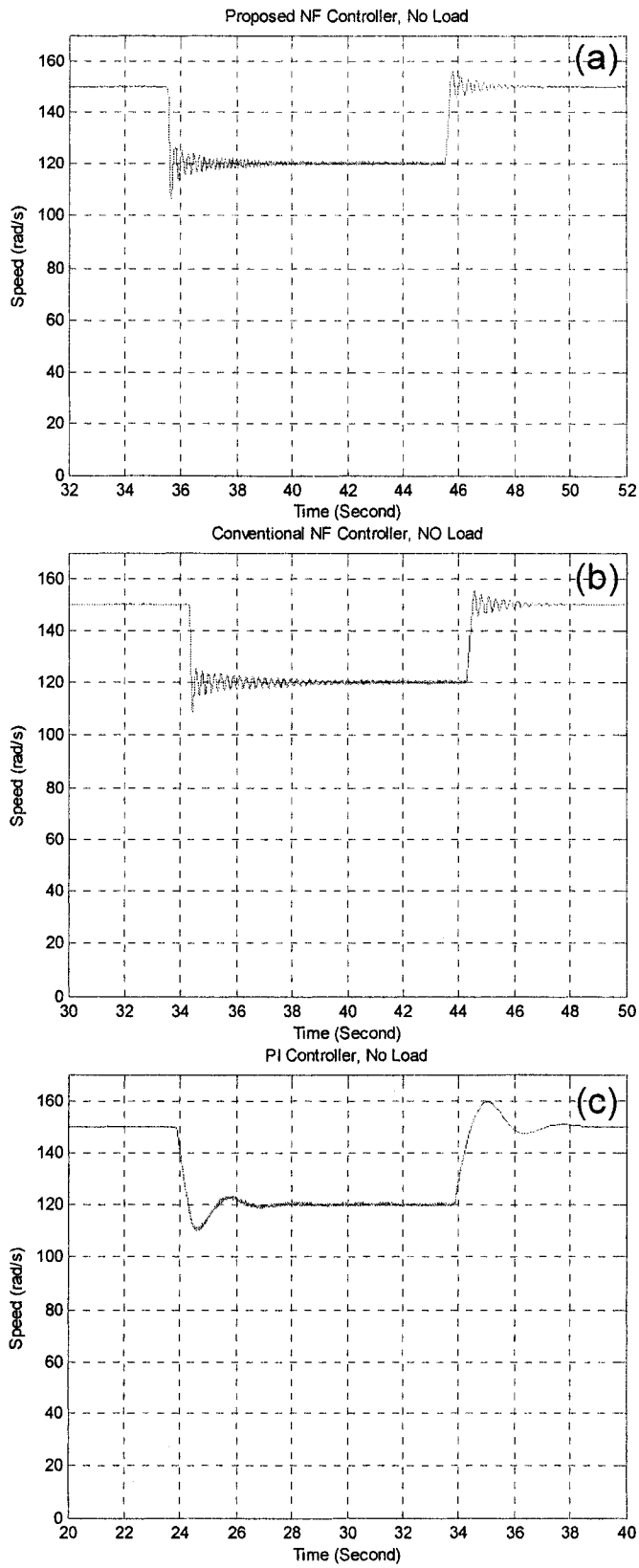


Fig. 3.17 Experimental speed response of the IM drive due to a step change in command speed at no load: (a) proposed NFC, (b) 2-input NFC, (c) PI.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

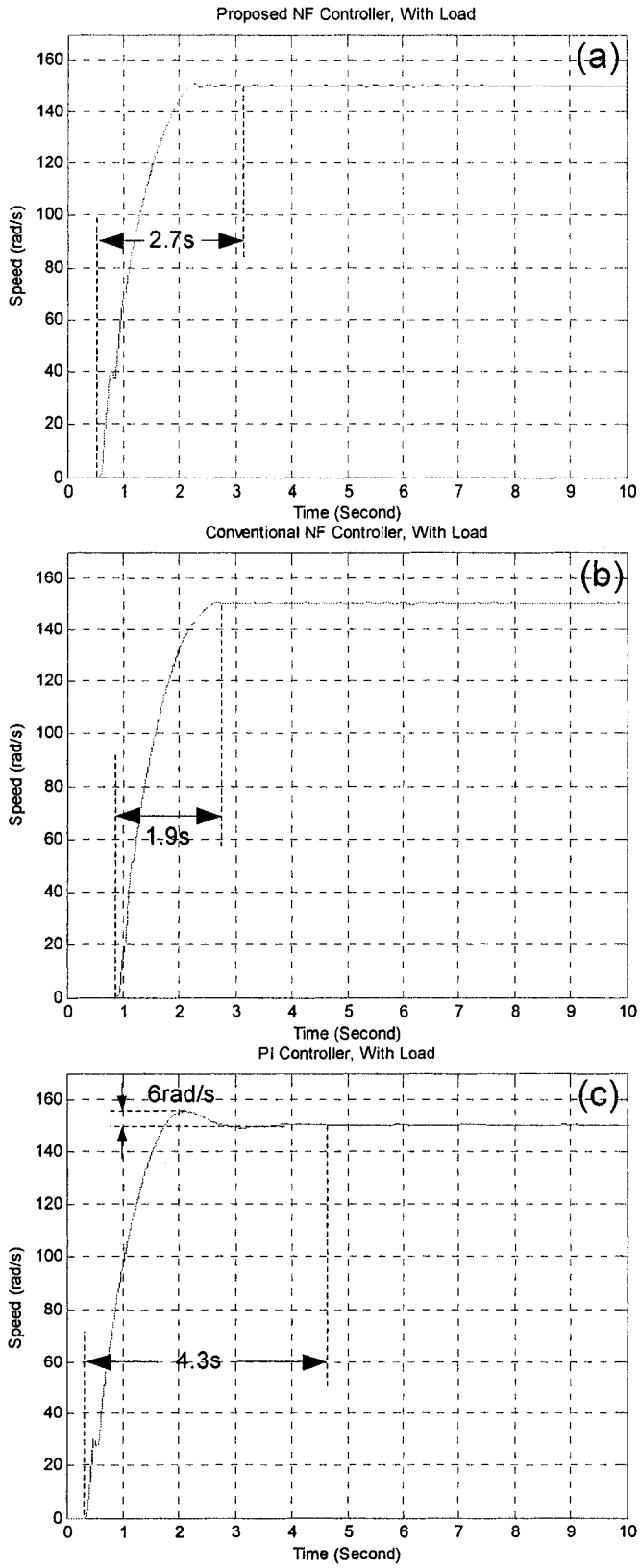


Fig. 3.18 Experimental starting speed responses of the drive at load condition: (a) proposed NFC, (b) 2-input NFC, (c) PI.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

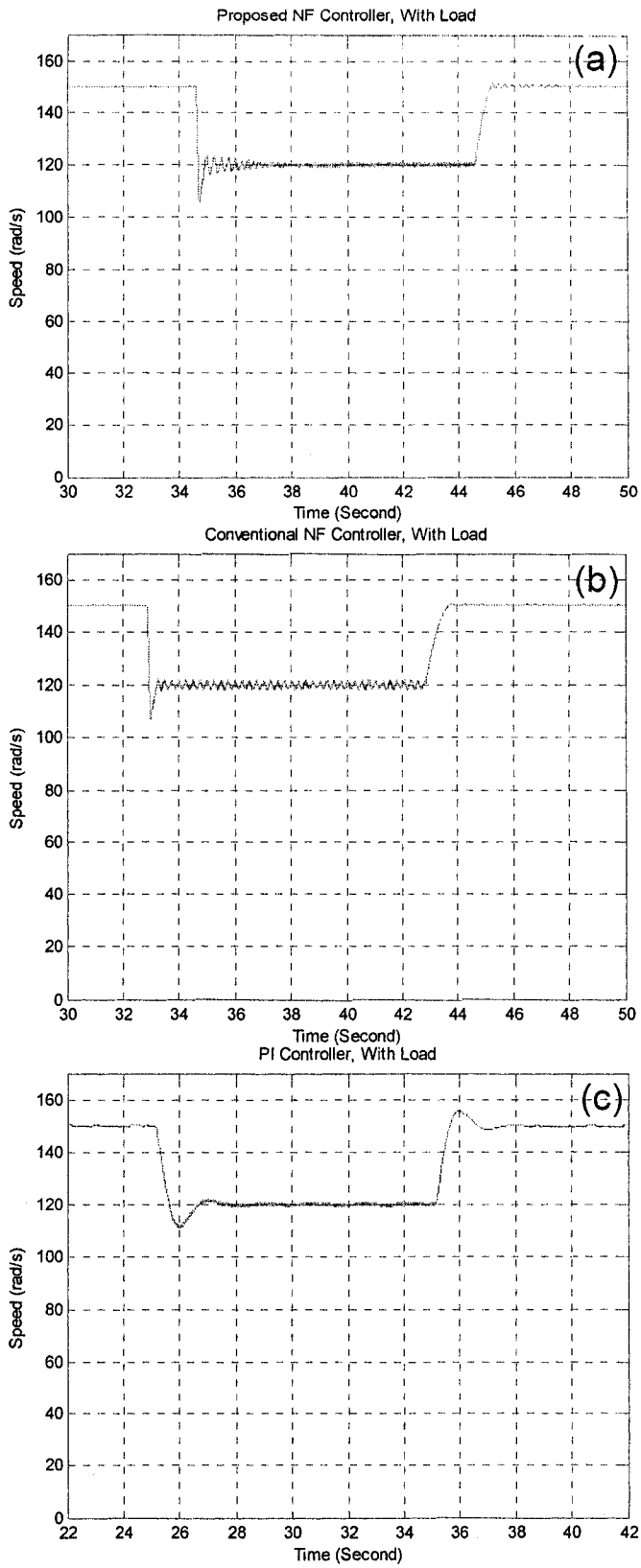


Fig. 3.19 Experimental speed response of the IM drive due to a step change in command speed at load condition: (a) Proposed NFC, (b) 2-input NFC, (c) PI.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

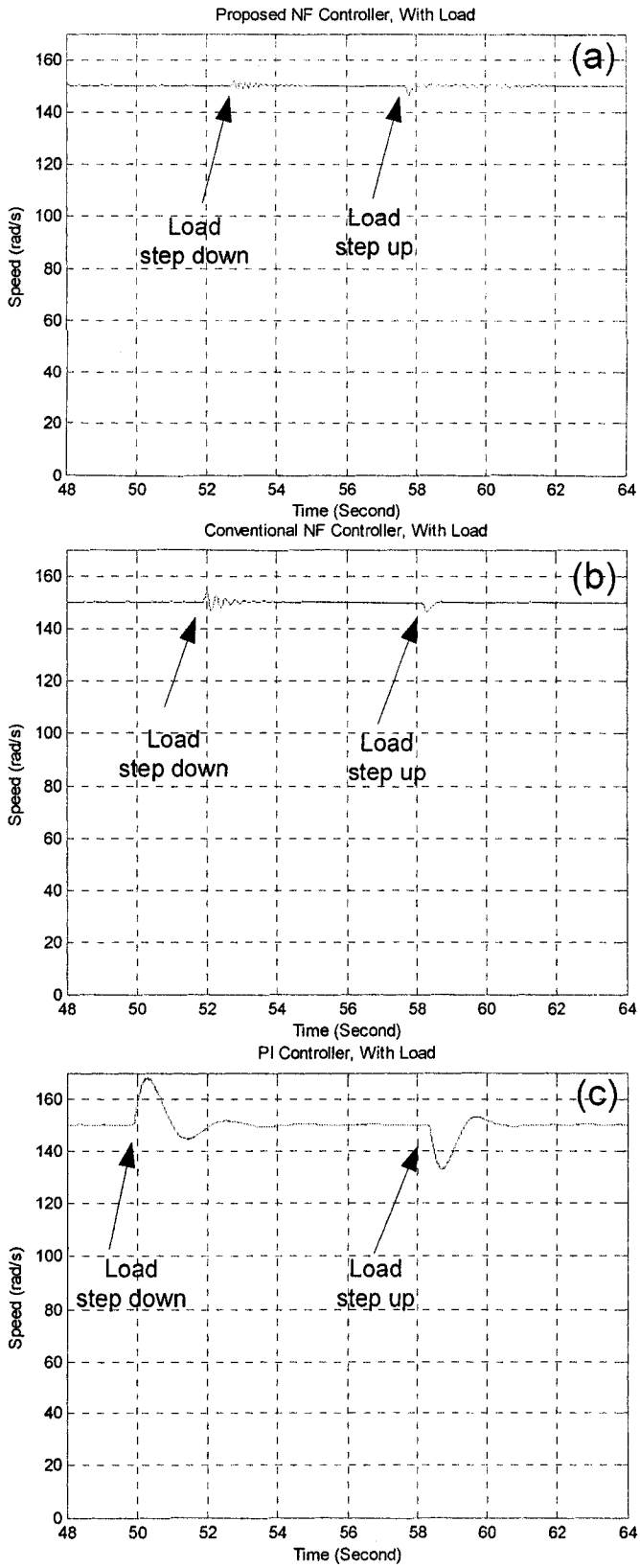


Fig. 3.20 Experimental performance of the IM drive due to a step change in load for: (a) proposed NFC, (b) 2-input NFC, (c) PI.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

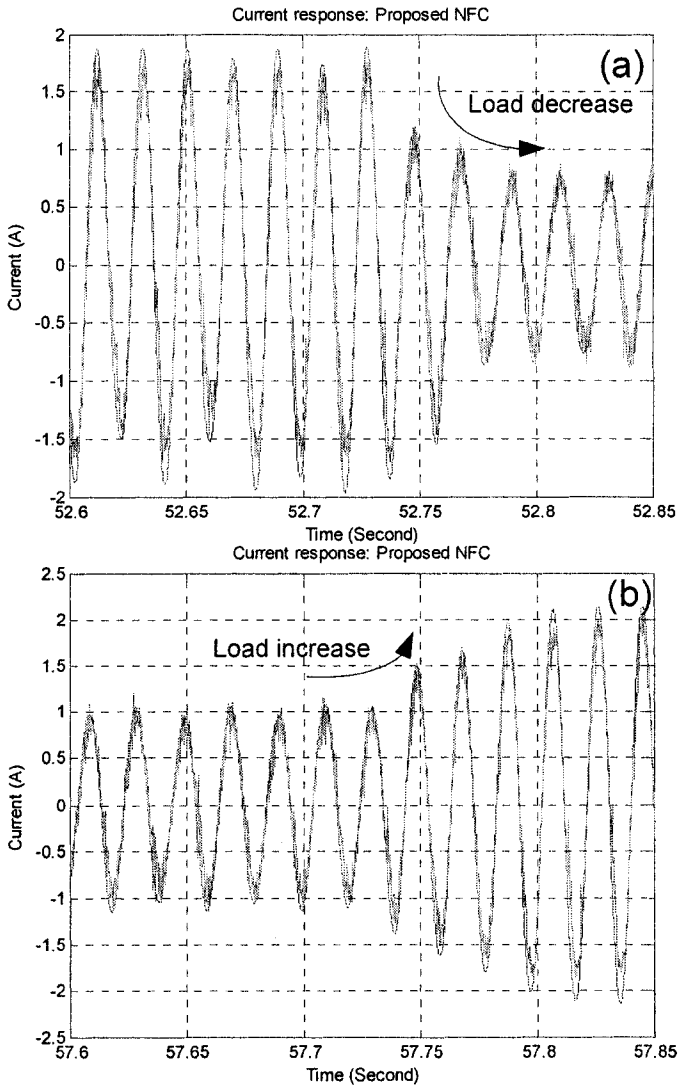


Fig. 3.21 Comparison of the reference and actual stator currents: (a) load decrease, and (b) load increase.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

3.6 Conclusion

In this chapter, a novel and low computational on-line self-tuning NFC-based speed control of IM drive has been developed and experimentally implemented for a laboratory 1/3 hp motor. In the proposed NFC, both weights and membership functions are on-line tuned based on operating conditions. The proposed controller can also be applied to other types of motors of different sizes only by adjusting the tuning rates. The comparison of the proposed NFC with a conventional 2-input NFC has also been presented in simulation. It is found that without any significant performance decreasing, the simplified structure reduces the computational burden and is easier to implement in real-time as compared to the conventional 2-input NFC. The proposed simplified self-tuned NFC-based IM drive system is found robust and could be a potential candidate for high performance industrial drive applications.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Chapter 4

A Novel Neuro-fuzzy Based Speed Ripple Minimization of Faulty Motor with Broken Rotor Bars

In this chapter a novel self–tuned neuro-fuzzy controller (NFC) is developed to minimize the speed ripple of a faulty induction motor with broken rotor bars (IMBRB). First the performance of an IMBRB is investigated in the open-loop condition. Then a new mechanical model of induction motor is presented and a new NFC is proposed to control the IMBRB in an indirect field oriented control scheme. The proposed NFC maintains the system performance by minimizing the speed ripples with electrical field frequency. Based on the knowledge of motor control and intelligent algorithms a supervised self-tuning method is also developed to adjust weights of the proposed NFC. The convergence/divergence of the weights is discussed and investigated by simulation. The complete drive is experimentally implemented using a digital signal processor board DS-1104 for a laboratory 1/2 hp

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

faulty motor. The effectiveness of the proposed NFC is tested both in simulation and experiment.

4.1 Open Loop Study of IMBRB

The faulty motor considered in this work has 3 broken rotor bars as shown in Fig. 4.1. The open loop experimental setup is shown in Fig. 4.2. The IMBRB was driven by a commercial v/f driver. The DC generator was working as a load to the IMBRB. The load level was set by adjusting the variable resistor. The objective of open loop study is to figure out the pattern and causes of speed ripples. Thus it will be helpful to develop a fault tolerant controller (FTC).

Theoretically, an IMBRB will display a low frequency of $2sf_e$ modulation in rotor speed (s is the rotor slip and f_e the supply frequency) [66], [89]. Moreover, the open loop experimental study found that the IMBRB also displays speed oscillation with electrical field frequency and its harmonics as shown in Fig. 4.3. Table 4.1 summarized open-loop experimental results at different load and speed settings. It can be seen that the magnitude of speed ripple with fundamental field frequency is approximately 10-20 times larger, and speed ripple with twice field frequency is approximately 6-15 times larger than that of the low frequency speed ripple.

The high frequency speed ripples may be caused by the unevenness of air gap and finite number of stator slots and rotor bars [79], [80], stator asymmetry (e.g. stator winding fault or unbalance power supply) [78], [89]. Although these high frequency speed oscillations are not caused by motor defect itself, they deteriorate system performance much more than the low frequency speed ripple since they have much bigger magnitude.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

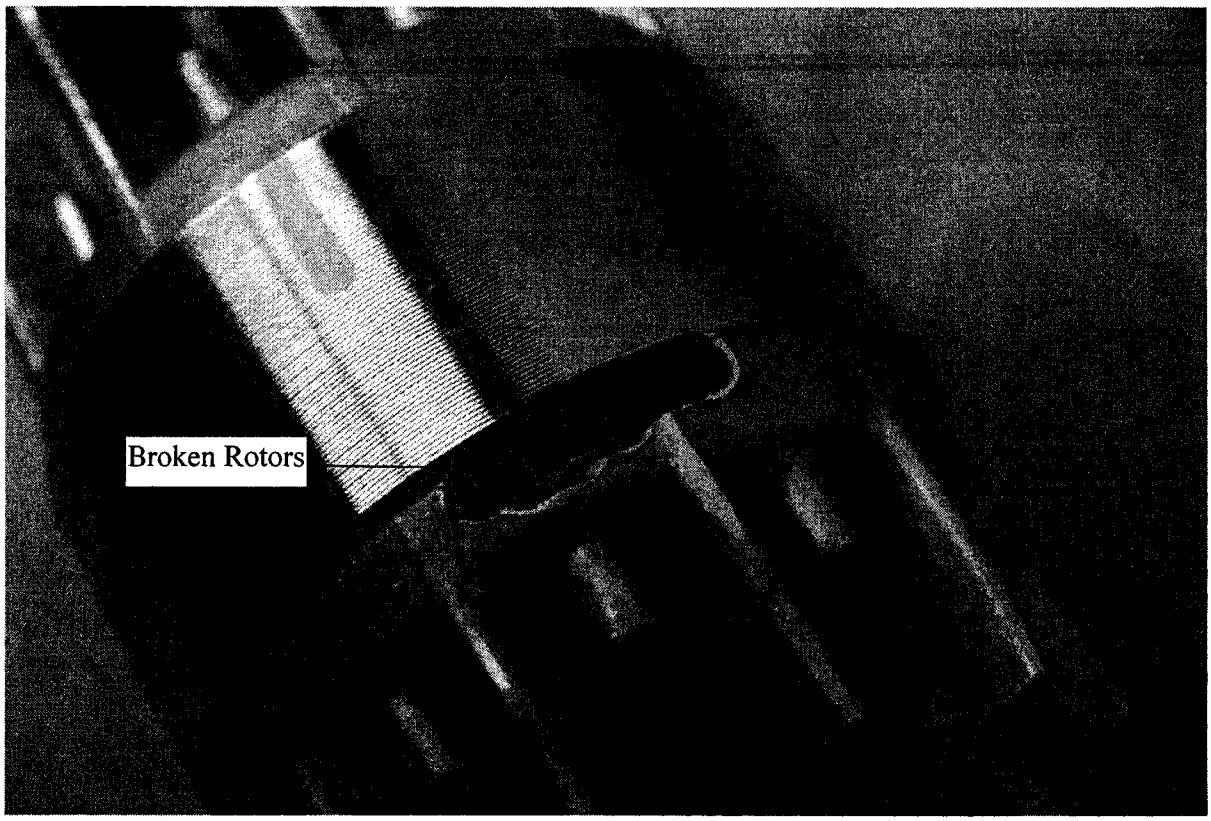


Fig. 4.1 The faulty rotors of the IMBRB.

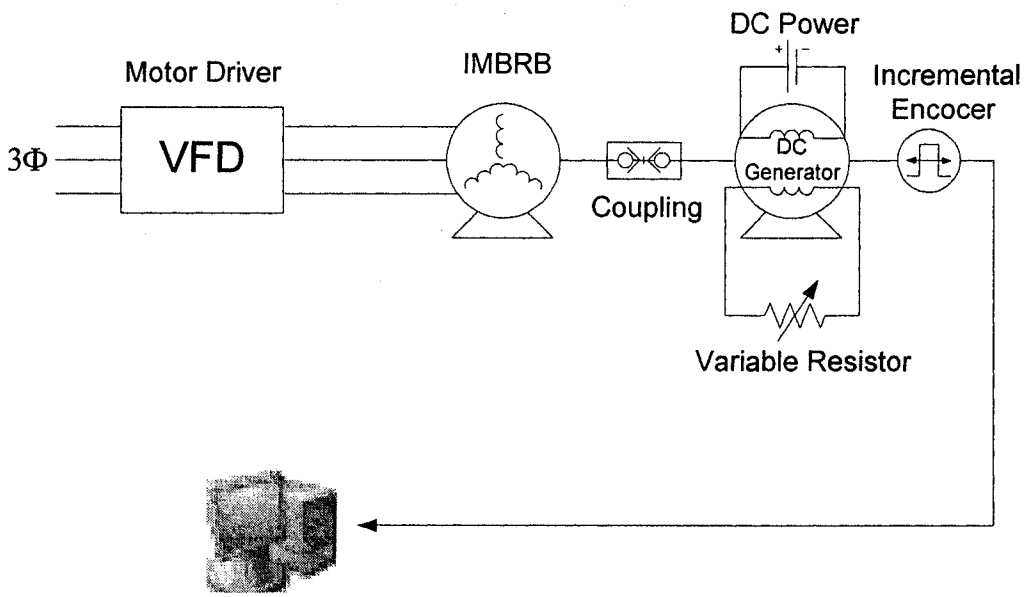


Fig. 4.2 Open loop experimental setup.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

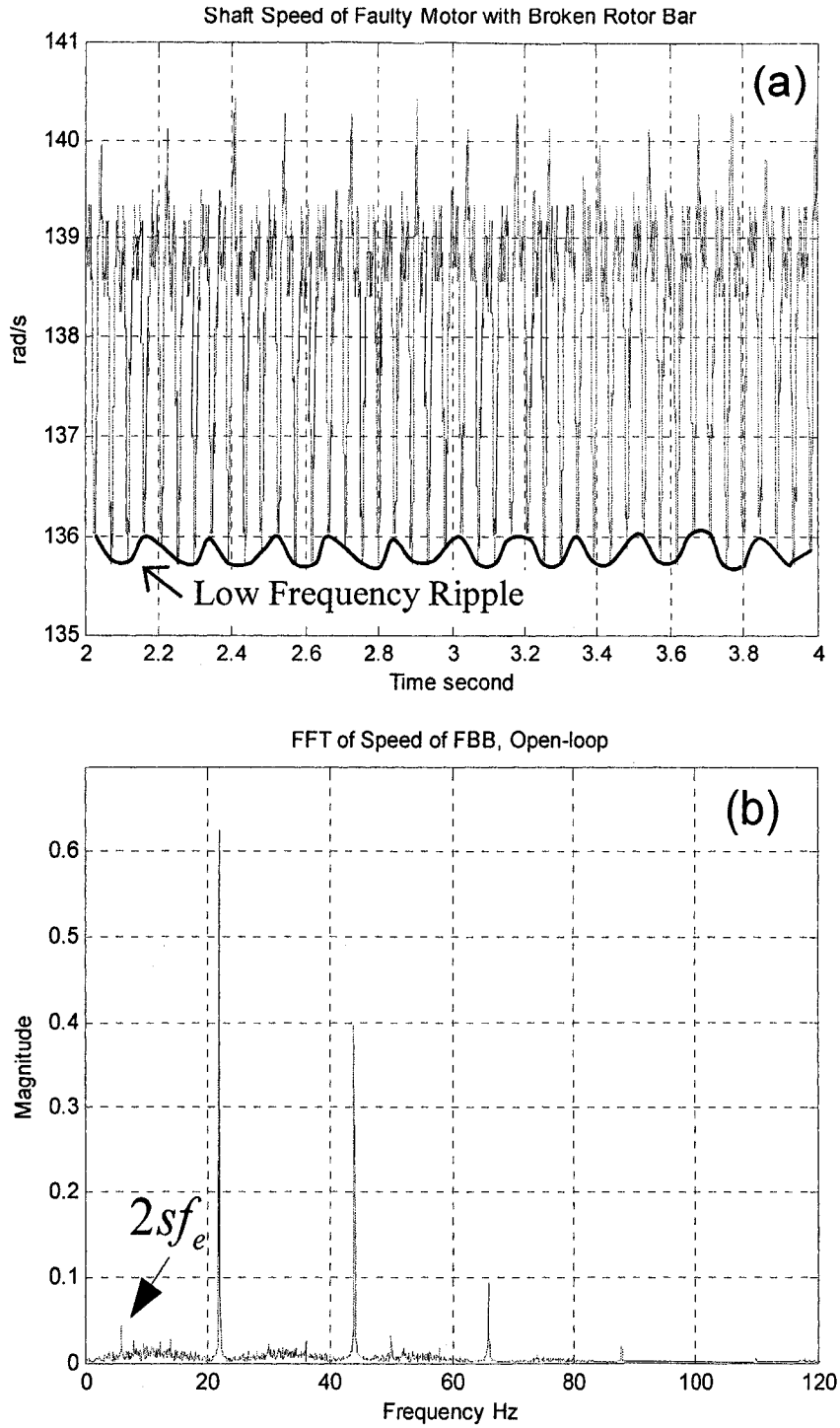


Fig. 4.3 (a) Rotor speed of an IMBRB, (b) FFT analysis of speed signal.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Table 4.1 FFT Analysis of Faulty Motor with Broken Rotor Bars

Load Level	Speed setting (rpm)	Speed Ripple		
		f_e (rad/s)	$2f_e$ (rad/s)	$2sf_e$
Low	600	0.266	0.252	0.014
↓	900	0.384	0.319	0.016
	1200	0.358	0.389	0.022
	1500	0.519	0.387	0.025
Medium	1800	0.733	0.381	0.033
Medium	600	0.214	0.211	0.021
↓	900	0.414	0.369	0.035
	1200	0.512	0.334	0.034
	1500	0.626	0.396	0.043
High	1800	0.740	0.434	0.070

Further analysis has been done to reveal the relationship between speed ripples and rotor position.

Fig. 4.4 (a) illustrates the relationship between the first field frequency speed ripple and the rotor position angle θ, while the Fig. 4.4 (b) the second field frequency speed ripple and 2θ. It gives us the theory support that the nth order speed ripple could be modeled by

$$\omega_{nth\_ripple} = K_a \cos(n\theta) + K_b \sin(n\theta) \tag{4.1}$$

The speed or torque ripple compensation methods for induction motors can be found in [66], [81], [90]. A compensation scheme found in [90] requires a torque

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

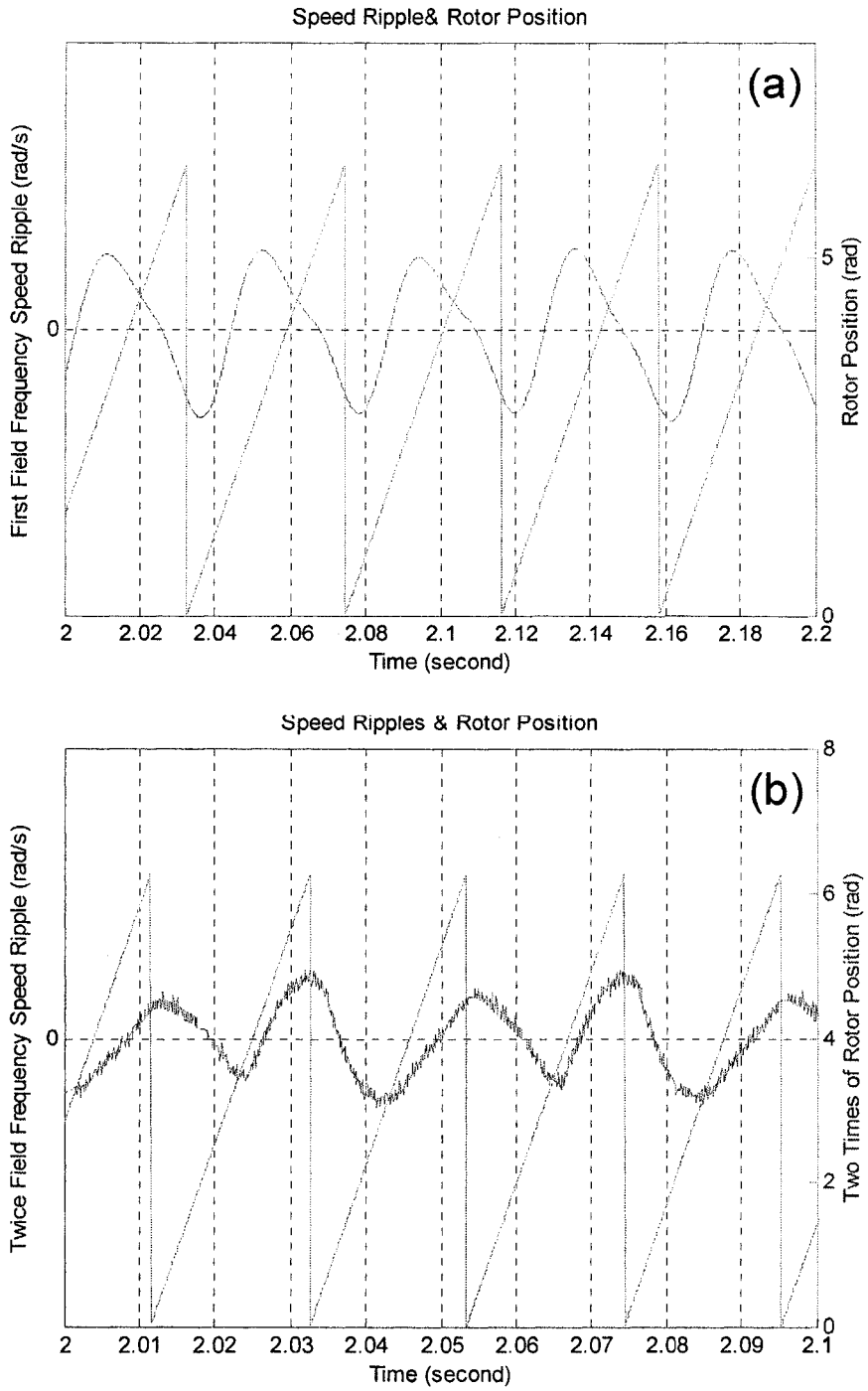


Fig. 4.4 (a) First field frequency speed ripple & rotor position, (b) Second field frequency speed ripple & rotor position.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

sensor to obtain a compensation torque, which maybe unacceptable in many application fields. In [81] the authors utilized a compensation method by correcting current measurement error. But the current measurement error is not the only source of speed ripples. In [66] the authors only mentioned and compensated the low frequency ripples of an IMBRB by using a fault-tolerant controller.

In this chapter a new NFC is developed to minimize speed ripples with field frequency and its harmonics which are significant components. The proposed NFC produces a constant torque plus a reverse ripple torque to fulfill two control targets: speed set point and lower speed ripple.

4.2 Mechanical System Model and Mathematical Analysis

The experimental observation suggests that:

1) The IM yields speed ripples under a constant torque;

2) The signature low frequency speed ripple $2sf_e$ can be neglected because of too small magnitude at the initial stage of fault;

3) The source of speed ripples can be modeled as a linear summation of sinusoidal functions whose frequencies are multiple of the field frequency.

Then the modified IM mechanical model can be proposed as follows,

$$\omega_m = \frac{1+\sum_{n=1}^{\infty}[K_{an}\cos(n\theta)+K_{bn}\sin(n\theta)]}{J_m s + B_m}(T_e^* - T_L) \tag{4.2}$$

where ω_m is rotor speed, J_m is moment of inertia, B_m is the coefficient of viscous friction, θ is rotor field angle and K_{an}, K_{bn} are scale factors representing magnitudes and phases of frequency components respectively. T_e^* is the desired developed torque, T_L is a constant load torque.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

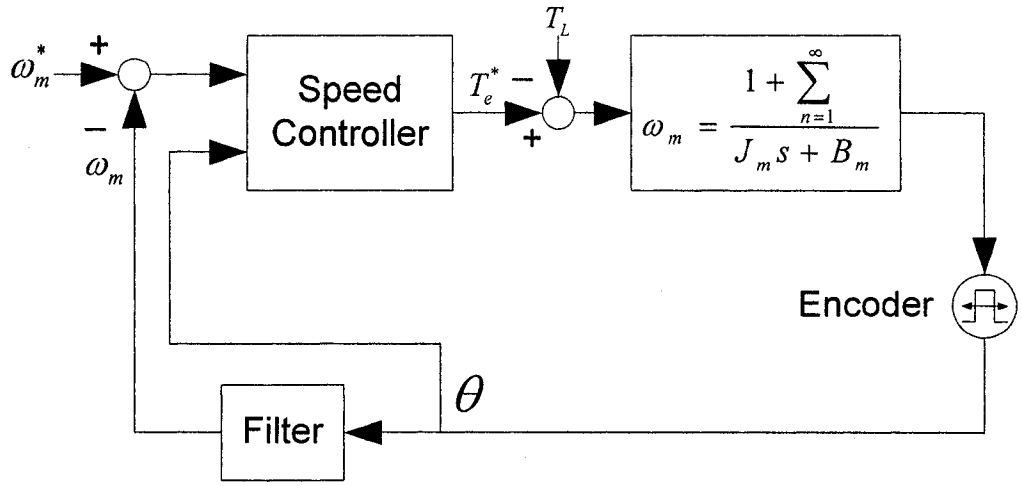


Fig. 4.5 IM control scheme.

Fig. 4.5 depicts the control scheme based on the proposed IM model.

From (4.2) one can get,

$$\begin{aligned} & J_m \frac{d\omega_m}{dt} + B_m \omega_m \\ & = T_e^* - T_L + (T_e^* - T_L) \sum_{n=1}^{\infty} [K_{an} \cos(n\theta) + K_{bn} \sin(n\theta)] \end{aligned} \tag{4.3}$$

In order to reach the reference speed setting as well as minimize the speed ripples, the developed torque can be decomposed into two parts: T_{ss} and T_{ripple}. T_{ss} is a constant torque regarding reaching the speed reference, and T_{ripple} is the reverse ripple torque regarding eliminating the speed ripples. Then the equation(4.3) can be written as following,

$$\begin{aligned} & J_m \frac{d\omega_m}{dt} + B_m \omega_m \\ & = \{T_{ss} - T_L\} + \left\{ (T_{ss} - T_L + T_{ripple}) \sum_{n=1}^{\infty} [K_{an} \cos(n\theta) + K_{bn} \sin(n\theta)] + T_{ripple} \right\} \end{aligned} \tag{4.4}$$

To eliminate the speed ripple, it means that:

$$\begin{aligned} & (T_{ss} - T_L + T_{ripple}) \sum_{n=1}^{\infty} [K_{an} \cos(n\theta) + K_{bn} \sin(n\theta)] + T_{ripple} = 0 \\ & T_{ss} - T_L = B_m \omega_m \end{aligned} \tag{4.5}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Then T_{ripple} and T_{ss} can be obtained by:

$$T_{ripple} = \frac{-(T_{ss} - T_L)\sum_{n=1}^{\infty}[K_{an}\cos(n\theta) + K_{bn}\sin(n\theta)]}{1 + \sum_{n=1}^{\infty}[K_{an}\cos(n\theta) + K_{bn}\sin(n\theta)]}$$
$$T_{ss} = B_m\omega_m + T_L \tag{4.6}$$

Equation (4.6) shows great complexity and nonlinearity. Traditional techniques are difficult to fit such kind of equation. However Neuro-fuzzy is a good candidate to approximate nonlinear functions, thanks to its ability to learn. Speed error $\omega_m^* - \omega_m$, rotor field angle θ are chosen to be the two inputs of the NF speed controller.

4.3 Design of the Proposed NFC for IMBRB

The proposed NFC incorporates fuzzy logic and a learning algorithm with a five-layer artificial neural network (ANN) structure as depicted in Fig. 4.6. The learning algorithm modifies the NFC to closely track the speed reference, and at the same time minimize the speed ripples. The detailed discussions on different layers of the NFC are given below.

Input Layer:

The normalized speed error, rotor field angle are the inputs of the proposed NFC, which are given by,

$$O_1^I = Input\ 1 = \frac{\omega_m^* - \omega_m}{\omega_m^*} * 100\%, \tag{4.7}$$

$$O_2^I = Input\ 2 = \theta, \tag{4.8}$$

where ω_m is the measured speed, ω_m^* is the command speed, θ is rotor field angle which can be measured by encoder, I denotes the 1st layer.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Fuzzification Layer:

In the proposed NFC, triangular and trapezoidal functions are chosen as the membership functions in order to lower the computational burden. For input 1, the membership functions are shown in Fig. 4.7. For input 2, the membership functions are in

Fig. 4.8. The output of the 1st layer is the input of the 2nd layer.

Rule Layer:

The *Rule Layer* includes *Rule If Layer* and *Rule Then Layer*. Based on our knowledge the rules are formed between inputs of O_1^I and O_2^I. Fuzzy singleton rules [74] are utilized in our proposed NFC as the following form,

$$R_1^j: \text{IF } O_1^I \text{ is } A_1^j \text{ AND } O_2^I \text{ is } A_2^j, \text{ THEN y is } w_j.$$

Rule If Layer:

The multiplication method is chosen to implement 'AND' Logic. The node equation in this layer is,

$$O_i^{III} = x_j^{III} x_k^{III}, \tag{4.9}$$

where x_j^{III} is the input of the 3rd layer which is same as the output of 2nd layer.

Rule Then Layer:

Sugeno zero mode is adopted in this proposed NFC, which utilizes crisp numbers instead of fuzzy numbers as the rule's results. The node equation in this layer is,

$$O_i^{VI} = x_i^{VI} w_i, \tag{4.10}$$

where x_i^{VI} is the input of the 4th layer which is same as the output of 3rd layer, w_i is the weight at *i*th node.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

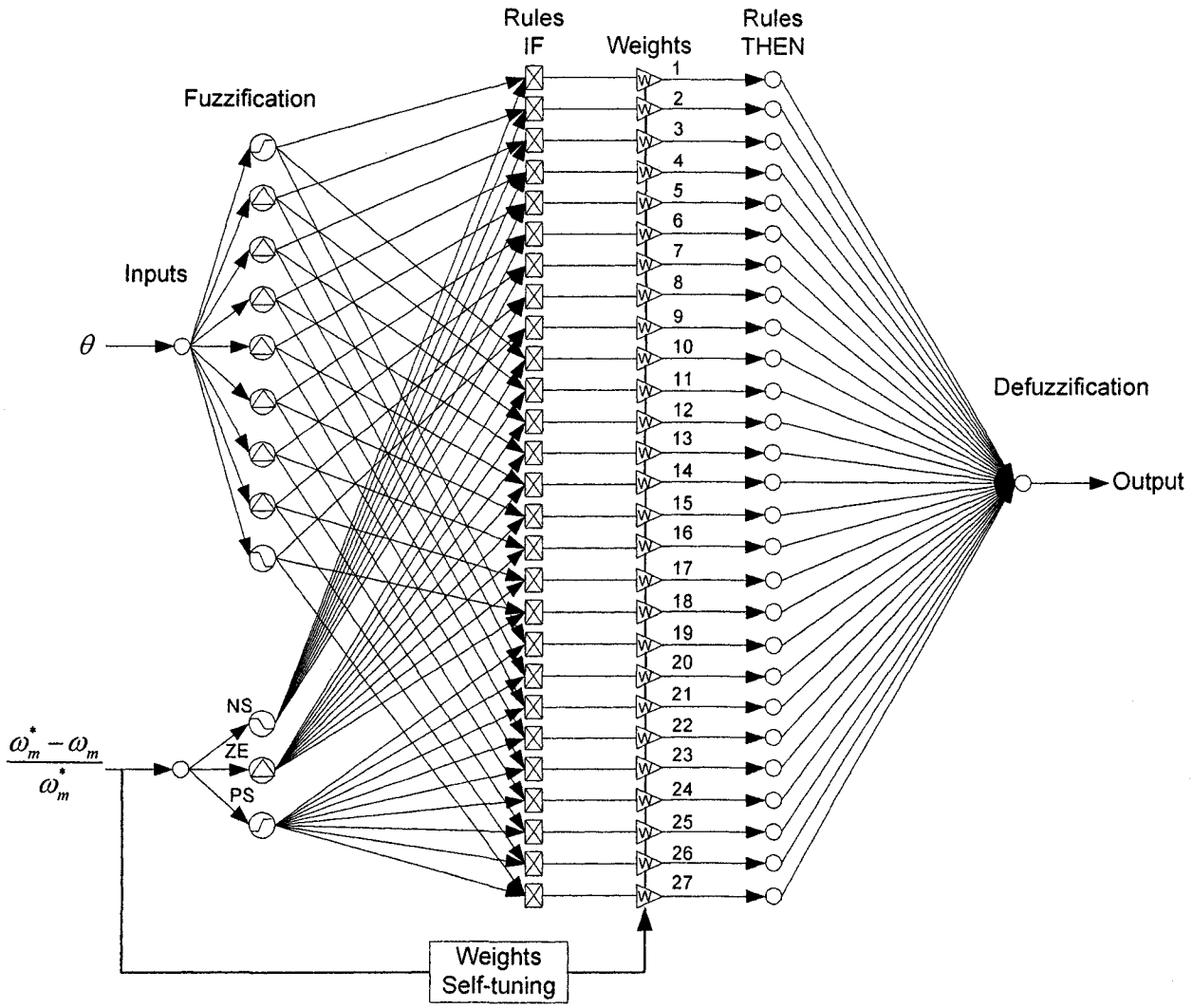


Fig. 4.6 Structure of the proposed NFC for IMBRB.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

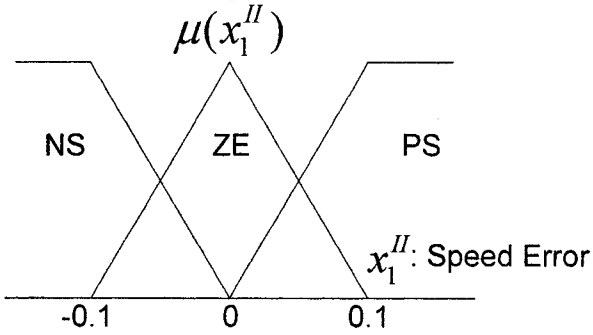


Fig. 4.7 Membership functions for input 1.

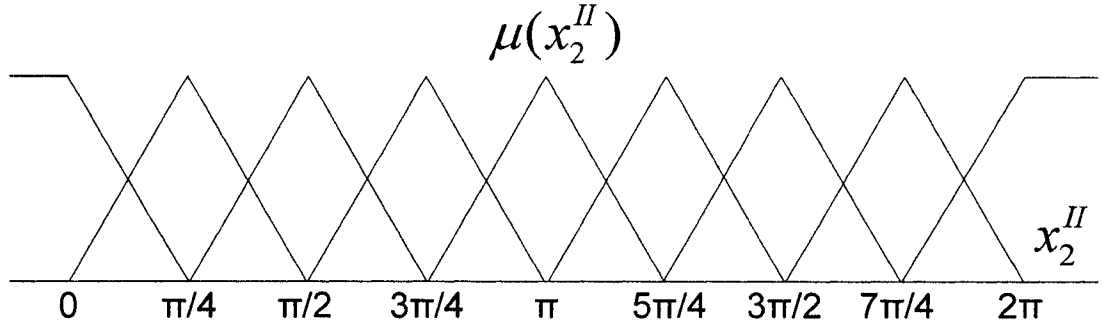


Fig. 4.8 Membership functions for input 2.

Defuzzification Layer:

The center of gravity method is used to determine the output of NFC. The node equation is specified as:

$$y = O_i^V = \frac{\sum x_i^V}{\sum O_j^{III}} = \frac{\sum O_i^{VI}}{\sum O_j^{III}}, \tag{4.11}$$

where x_i^V is the input of the 5th layer which is same as the output of 4th layer.

4.4 On-Line Self-Tuning Algorithm

A supervised on-line self-tuning method is introduced in this work. The objective of the proposed NFC is to generate the desired command torque T_e^* in the

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

indirect field orientation, and then produce correct torque to counteract torque ripples. The difficulty is that the NFC output, which is needed for updating the weights, is not readily available but has to be induced from the known desired system output error.

Some methods have been proposed in the literatures to solve this problem [74].

1) Approximations based on a mathematical model of the system
2) Identification of system inverse dynamic
3) Reinforcement learning

The methods 1) and 2) are to derive the NFC output error from the system error measurements. In our case, for method 1) the system parameters have to be known, such as moment of inertia and viscous friction coefficient. The method 2) and 3) will bring heavy computational burden and complex algorithm.

In this work, the system output error is employed to guide the control action in the right direction to achieve desired response. The NFC parameters are directly tuned to reduce the output error.

When using the back propagation for control application, the weights are converging rather slowly [74]. In this paper the *Kaczmarz's Projection Algorithm* [91] is used to update the weights since it is faster than the BP algorithm.

The update rules are given as follows:

$$w_j(n) = w_j(n-1) + \gamma(u_d^{(n)} - u^{(n)}) \frac{O_j^{III}(n-1)}{\sum (O_j^{III} \wedge 2)}, \tag{4.12}$$

Where $u_d^{(n)} - u^{(n)}$ is the controller output error, $\gamma > 0$ is a learning factor. In terms of the Jacobean Matrix *J(n)* of the system, the system error and controller output error can be expressed as

$$u_d^{(n)} - u^{(n)} = \frac{1}{J(u)}(\omega_m^* - \omega_m). \tag{4.13}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

In equation (4.13) the Jacobean matrix *J(n)* is not easily found directly. As mentioned earlier in Chapter 3, IM can be viewed as one SISO system, then the *J(n)* can be estimated as a constant value $K_J > 0$ [76].

Then the equation (4.12) can be rewritten as,

$$w_j(n) = w_j(n-1) + \eta(\omega_m^* - \omega_m)\frac{O_j^{III}(n-1)}{\sum(O_j^{III} \wedge 2)} \tag{4.14}$$

where $\eta = \frac{\gamma}{K_J}$.

However this weight-updating method may cause problems. For the weights w_{1-9}, the updating is always happening at the time when $\omega_m^* - \omega_m \le 0$. Since $\eta > 0$ and $\frac{O_j^{III}(n-1)}{\sum(O_j^{III} \wedge 2)} \ge 0$, the weights w_{1-9} are consistently decreasing. On the contrary, the weights w_{19-27} are consistently increasing.

In this thesis an algorithm is developed to fix the problem, which consists of following three steps,

Step 1: Assign appropriate initial values to the weights w_{1-9} and w_{19-27}.

Since the weights w_{1-9} relate to the time when $\omega_m \ge \omega_m^*$, the NFC is desired to produce T_e^* which is smaller than the T_L, while weights w_{19-27} relate to the time when $\omega_m \le \omega_m^*$, the NFC is desired to produce T_e^* which is bigger than the T_L, the initial values are given as,

$w_{1-9} = -PB$, $w_{19-27} = +PB$, PB is a positive constant.

Step 2: Tune the weights w_{1-9} and weights w_{19-27} at different specific time as described below.

The IM mechanical equation is as following:

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$J_m \frac{d\omega_m}{dt} + B_m \omega_m = T_e^* - T_L . \tag{4.15}$$

When the system reaches the steady state, the speed is varying around the set point. At this time

$$\omega_m = \omega_m^* + \Delta\omega_m . \tag{4.16}$$

Meanwhile one can split the NFC output T_e^* into two parts: T_e^{ss} and T_e^{ripple}, and load torque into: T_L^{ss} and T_L^{ripple}. T_e^{ss} and T_L^{ss} are constant values, and T_e^{ripple} and T_L^{ripple} are ripple components which are varying continuously.

For a small change in torque ΔT, there is a corresponding change in speed $\Delta\omega_m$. Hence (4.15) can be rewritten as,

$$J_m \frac{d\Delta\omega_m}{dt} + B_m \Delta\omega_m + B_m \omega_m^* = \Delta T + T_e^{ss} - T_L^{ss}, \tag{4.17}$$

where $\Delta T = T_e^{ripple} - T_L^{ripple}$. Obviously, $B_m \omega_m^* = T_e^{ss} - T_L^{ss}$. Thus (4.17) is reduced to

$$J_m \frac{d\Delta\omega_m}{dt} + B_m \Delta\omega_m = \Delta T \tag{4.18}$$

From (4.18) one can see that,

$$\frac{d\Delta\omega_m}{dt} \le 0 \;\&\; \Delta\omega_m \le 0 \,,\; \Delta T \le 0 \,;$$

$$\frac{d\Delta\omega_m}{dt} \ge 0 \;\&\; \Delta\omega_m \ge 0 \,,\; \Delta T \ge 0 \,.$$

$\Delta\omega_m$ is the system error and ΔT is the controller output error. This gives the specific time when the weights w_{1-9} and weights w_{19-27} will be tuned as illustrating in Fig. 4.9. Since speed fluctuation happens at the same rotor position at different time, w_{1-9} / w_{19-27} are decreasing / increasing at different rate, respectively.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Step 3: Stop tuning when the system error is smaller than the threshold.

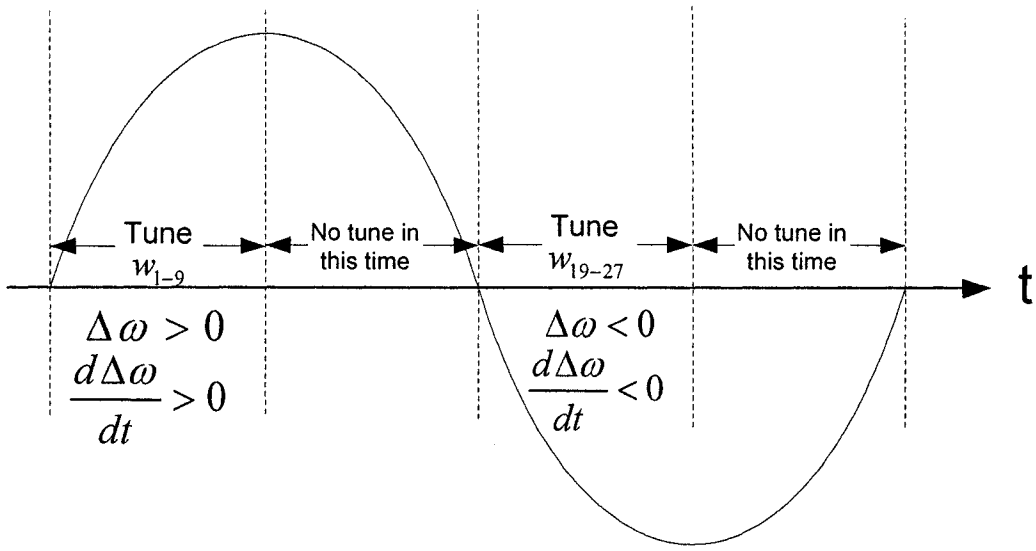


Fig. 4.9 Weights w_{1-9} and w_{19-27} tuning time.

4.5 Simulation Study

A simulation model of indirect FOC of IMBRB drive is developed in Matlab/Simulink[72] according to Fig. 4.10. The electrical and mechanical parameters are listed in Appendix A-2. The IMBRB has been damaged by introducing a rotor fault as shown in Fig. 4.1. The total load torque is added in as follows,

$$T_L = T_{Lapp} + T_{Ldist} \tag{4.19}$$

$$T_{Lapp} = B_m \omega_m \tag{4.20}$$

$$T_{Ldist} = \sin(\theta) + 0.5\sin(2\theta) \tag{4.21}$$

where T_{Lapp} is applied load torque, T_{Ldist} is the added disturbance to the load torque and θ is rotor angle.

The performance of the proposed NFC is compared to a fine tuned PI controller at different reference speed. To make a fair comparison, the PI controller is

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

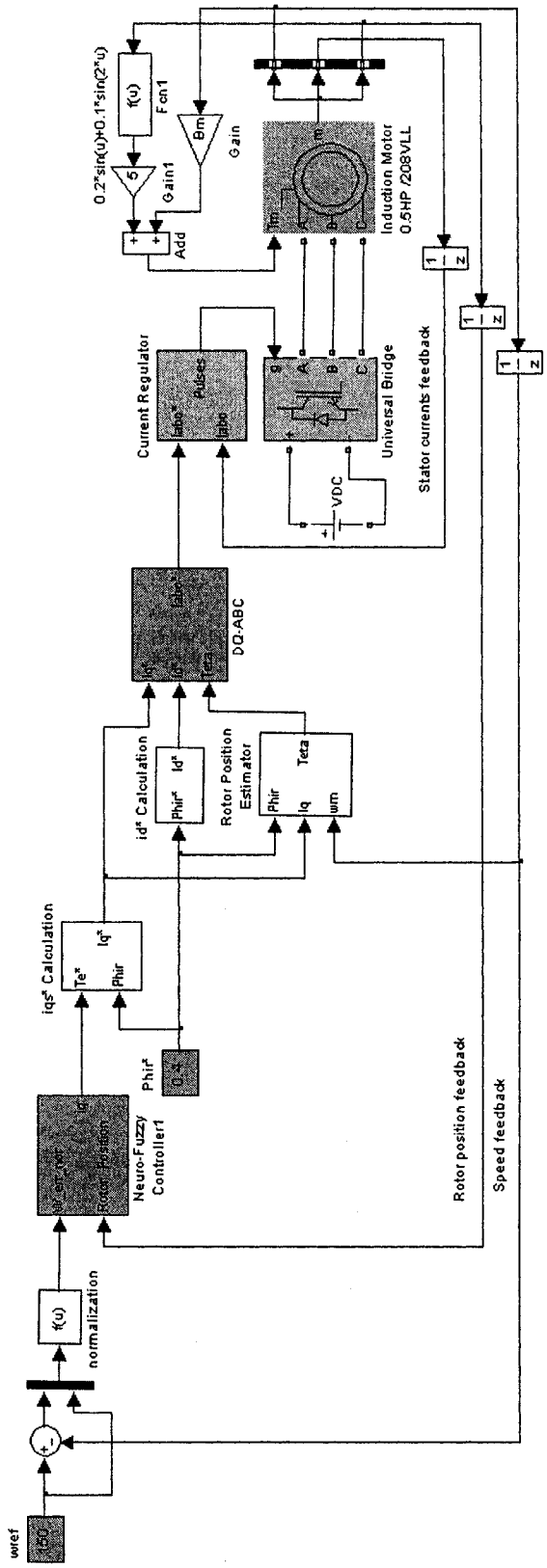


Fig. 4.10 Block diagram of the proposed NFC based speed ripple minimization of IMBRB drive.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

readjusted whenever reference speed is changed and the P-gain is set as large as possible. The parameters of the PI and NFC are listed as following.

PI: $K_p = 0.9,\ K_i = 0.6 \quad \omega_m^* = 100 rad/s$

$K_p = 0.5,\ K_i = 0.3 \quad \omega_m^* = 150 rad/s$

$K_p = 0.4,\ K_i = 0.3 \quad \omega_m^* = 180 rad/s$

NFC: *Learning Rate = 0.1*

Initial Value $w_{1\sim9} = -2.5$, $w_{19-27} = +2.5$

Fig. 4.11-4.13 show the simulated comparative speed response between PI and the proposed NFC based IMBRB drive at speed references 100rad/s, 150rad/s, and 180rad/s, respectively. It is evident that the proposed NFC significantly reduced peak-to-peak speed ripples at different operation conditions as compared to conventional PI controller.

The weights w_{10-18} are crucial to the effectiveness of the speed ripple minimization and system stability. The Fig. 4.14-Fig. 4.16 show the trend of the weights w_{10-18} changing at speed references 100rad/s, 150rad/s, and 180rad/s, respectively. The simulation results suggest that the weights w_{10-18} are convergent at different values. It proves the effectiveness of the proposed tuning algorithm and stability of the proposed NFC.

One simulation was conducted at reference speed of 150rad/s considering the speed loop delay and quantization error introduced by the encoder in order to simulate the real-time situation. Instead of directly obtaining the rotor speed from the IM model, in this simulation the speed is calculated by

$$\omega_m(n) = \frac{\theta(n) - \theta(n-10)}{10T_S}. \tag{4.22}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

where θ is the rotor position and T_S is the sampling interval. The quantization error is simulated by a set of embedded code in Matlab/Simulink. Fig. 4.17 compares the speed response of PI and proposed NFC considering the speed loop delay of $10Ts$ and quantization error. It is proved that the proposed NFC shows better performance than PI controller in case of simulating a real-time situation.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

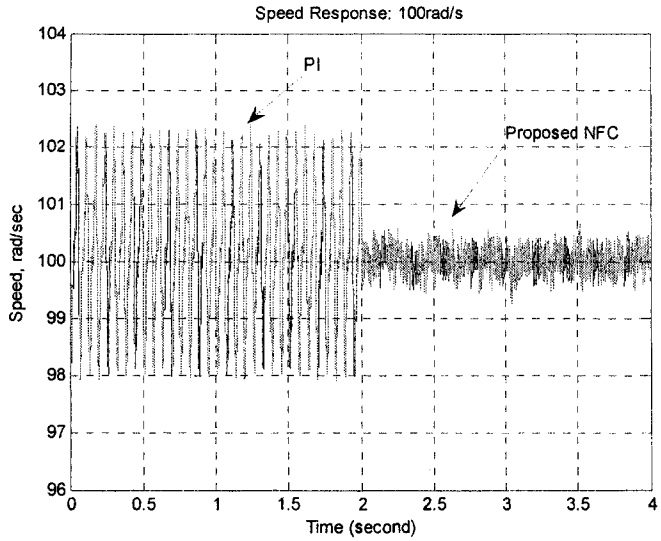


Fig. 4.11 Simulated speed response: reference 100*rad/s* (steady-state zoom in view).

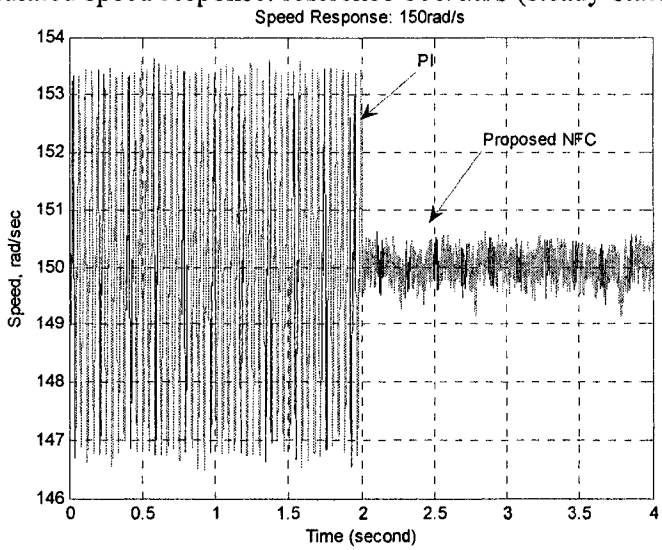


Fig. 4.12 Simulated speed response: reference 150*rad/s* (steady-state zoom in view).

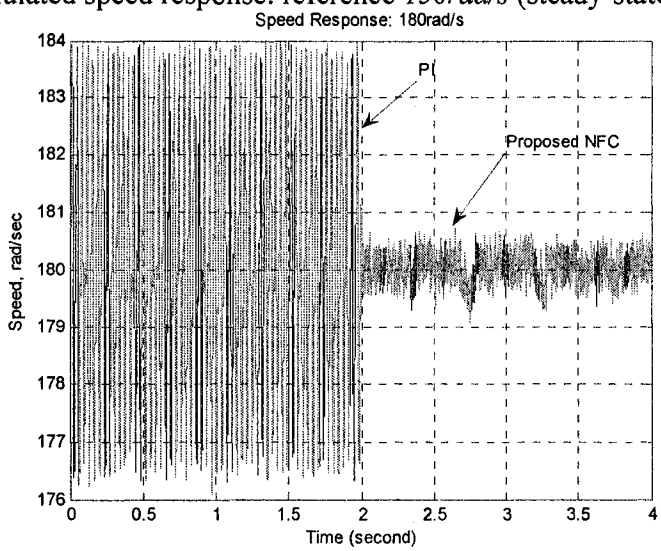


Fig. 4.13 Simulated speed response at 180*rad/s* (steady-state zoom in view)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

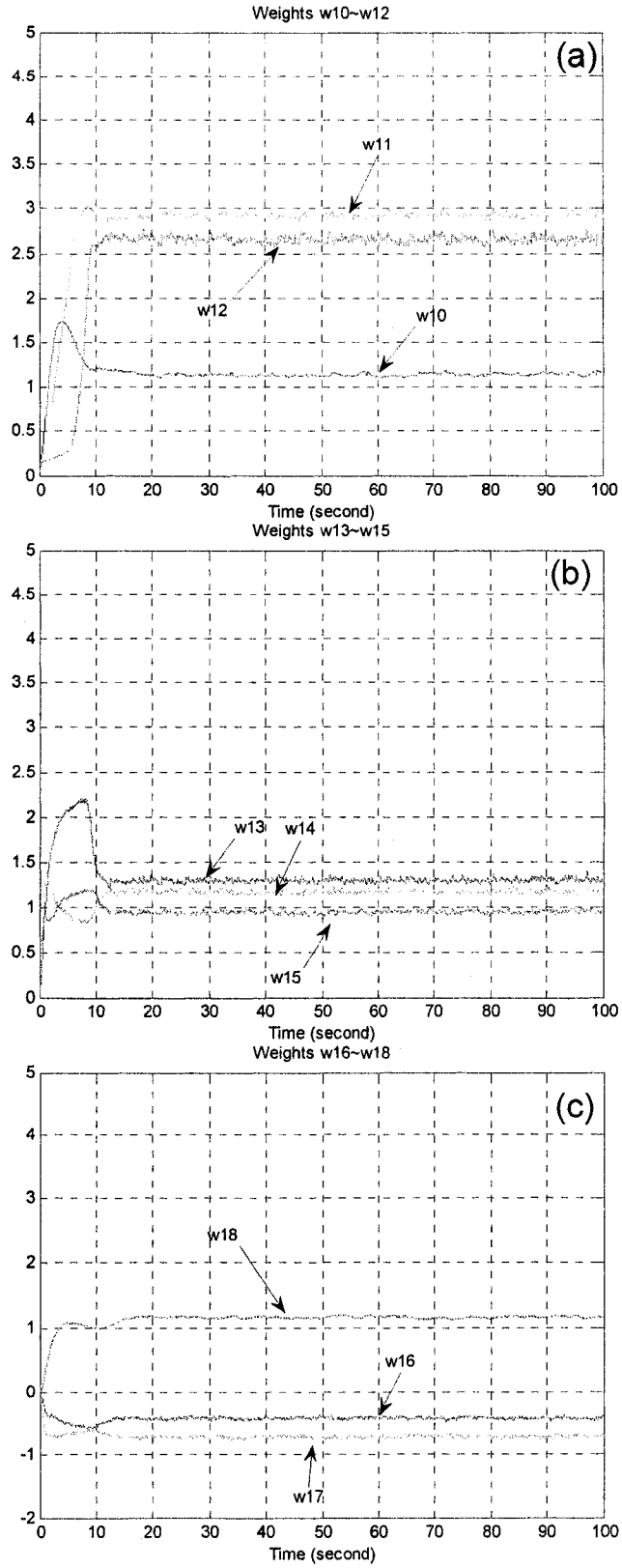


Fig. 4.14 Simulated results of weights variation at reference speed 100*rad/s*: (a) Weights 10-12, (b) Weights 13-15, (c) Weights 16-18.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

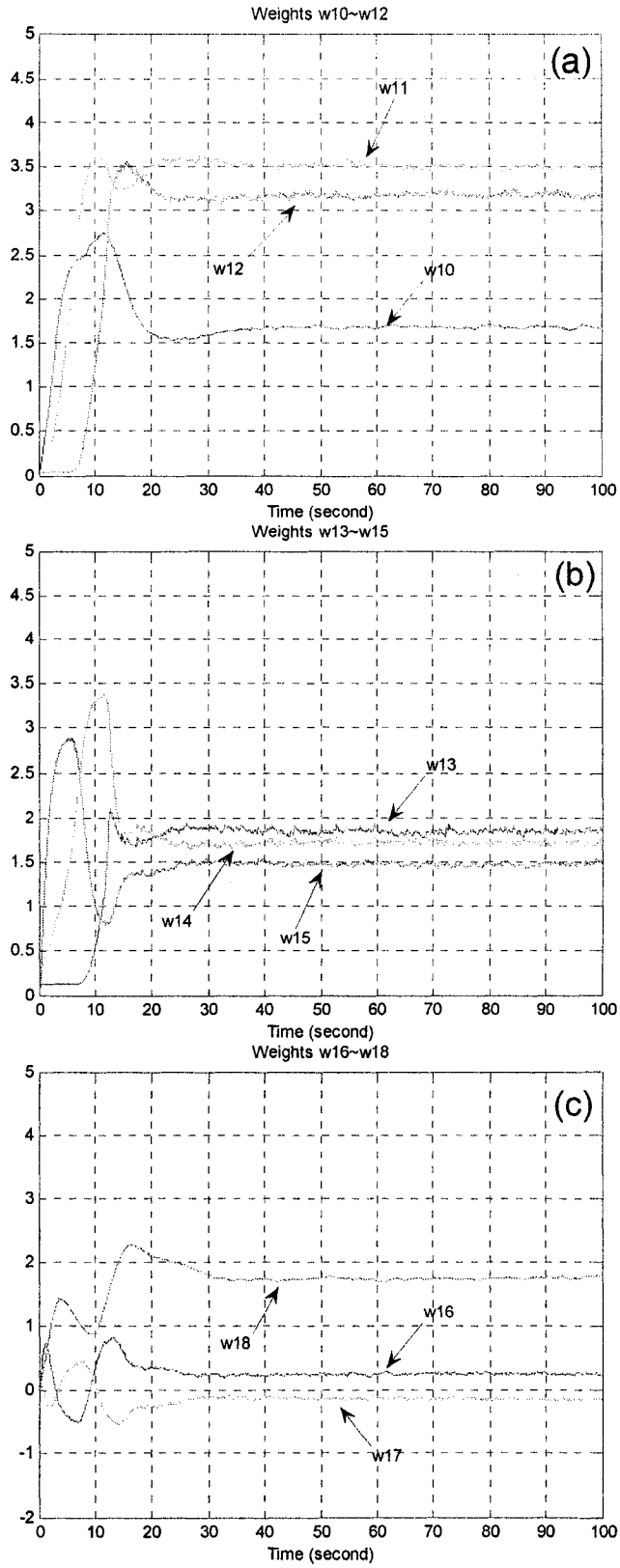


Fig. 4.15 Simulated results of weights variation at reference speed 150*rad/s*: (a) Weights 10-12, (b) Weights 13-15, (c) Weights 16-18.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

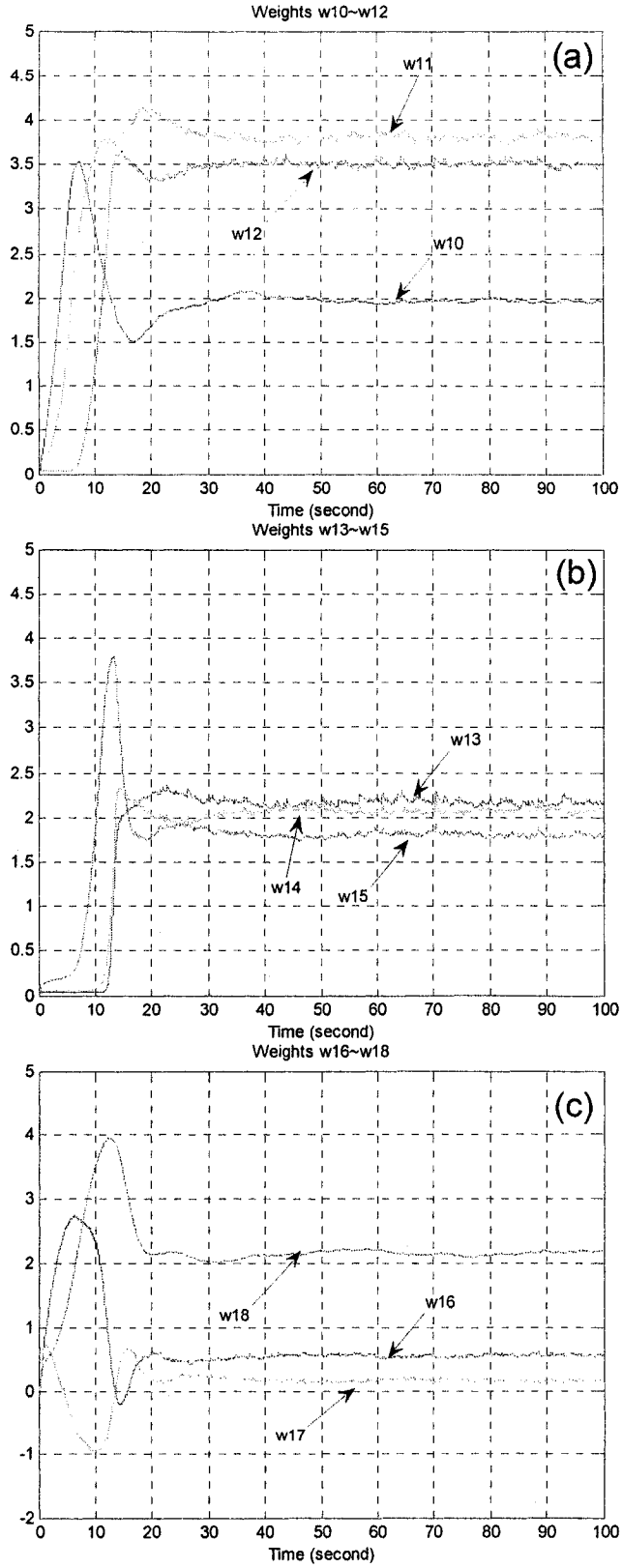


Fig. 4.16 Simulated results of weights variation at reference speed 180*rad/s*: (a) Weights 10-12, (b) Weights 13-15, (c) Weights 16-18.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

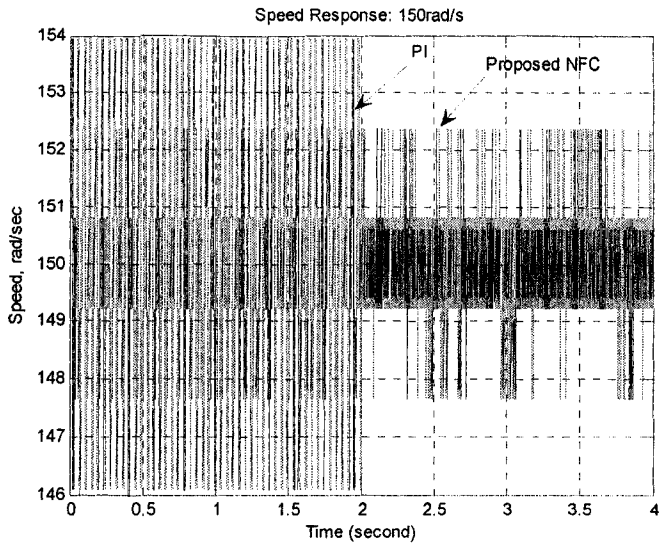


Fig. 4.17 Simulated speed response at reference speed 150*rad/s* considering the speed loop delay and quantization error (steady state, zoom in view)

4.6 Experimental Study

4.8.1 Factors Affecting Speed Ripple In Real-Time

Some factors influencing efficiency of speed ripple minimization include:

(1) Accuracy of speed measurement

(2) Delay in speed loop

The overall performance of a closed-loop system depends, to some extent, on the quality of the feedback signal which is the angular velocity in this case. Two types of velocity transducer are mostly used for high performance motor control systems: tachometer and optical encoder. But both have its drawbacks. The tachometer produces a DC voltage which is proportional to the shaft speed. The accuracy of a tachometer is suffering from its inherent non-linearity, temperature variations and components aging [81]. The encoder generates pulses representing rotor position which is essential in AC motor vector control system to implement field orientation. Velocity is estimated from discrete position data by approximate algorithms such as

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

derivation. Although several algorithms have been developed [82]-[87], it seems the estimate error and delay are inevitable.

In the experiment an incremental encoder with 1000 pulse per revolution is used. The rotor speed is obtained by counting the number of pulse in a fixed period mT_s.

$$\omega_m = \frac{N(n) - N(n-m)}{mT_s}$$

where T_s is the sampling time. Because the pulse counting is actually a rising edge counting and the start point of this fixed period is random, a quantization error is produced which is illustrated in Fig. 4.18.

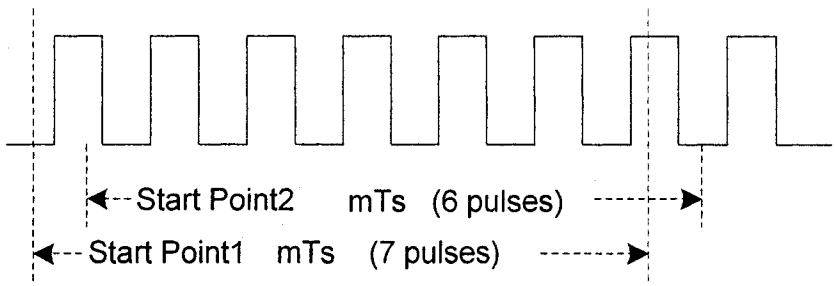


Fig. 4.18 Quantization error illustration

In case 1, the pulse number is counted as 7; in case 2, the pulse number is counted as 6, although the situation is same. The maxim value quantization error is $\frac{1}{mT_s} * \frac{2\pi}{4000}$.

The quantization error belongs to random error. Obviously this error will decrease as the time period increases. However, as the time period increases the speed loop feedback delay increases, the system dynamic response decreases. In order to select a proper time period for speed loop, an experiment was done using a PI controller. The m is set to be 1,5,10 and 100. Also in order to evaluate the system performance under different delay times, we collected the data of rotor position θ and calculated the rotor speed off-line by

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$\omega_m = \frac{\theta(n) - \theta(n-100)}{100T_s}. \tag{4.23}$$

Fig. 4.19-4.22 show the calculated speed response. And from the comparison of these experimental results, we finally chose *m=10* in our next experiments.

m	Quantization Error (rad/s)	Delay (second)
1	15.71	0.0001
5	3.14	0.0005
10	1.571	0.001
100	0.157	0.01

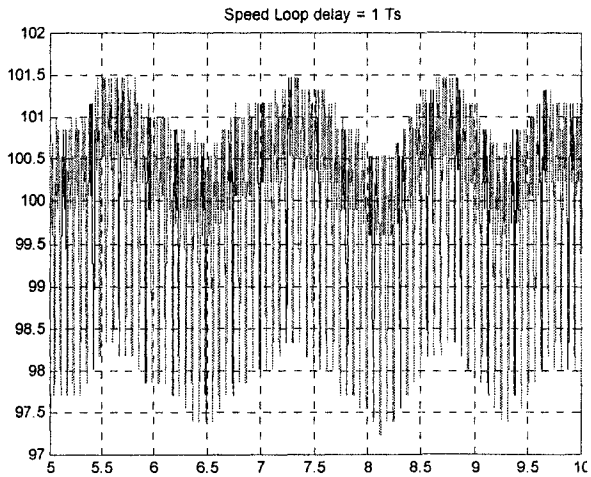


Fig. 4.19 Experimental result for m=1.

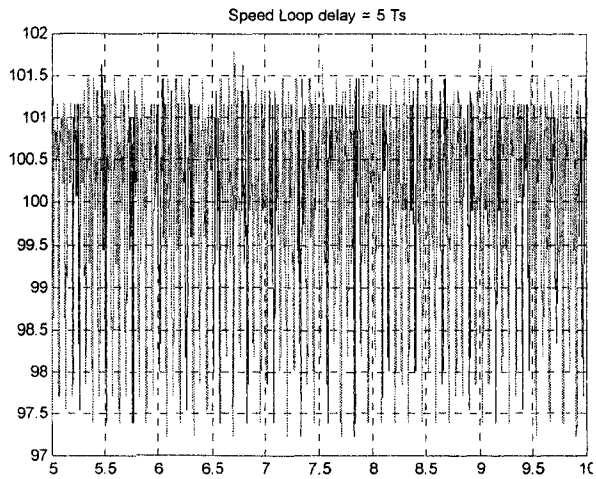


Fig. 4.20 Experimental result for m=5.

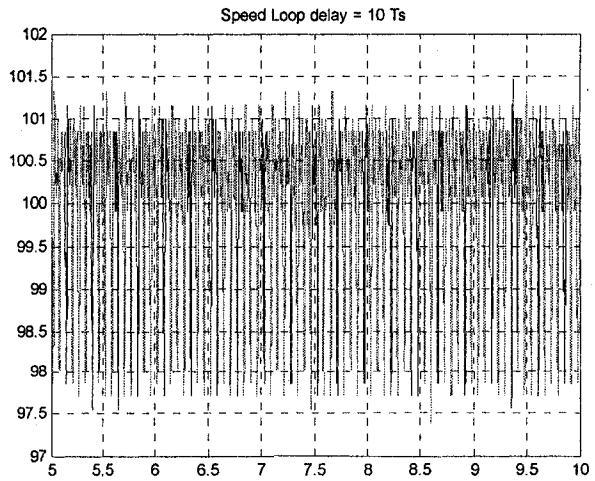


Fig. 4.21 Experimental result for m=10.

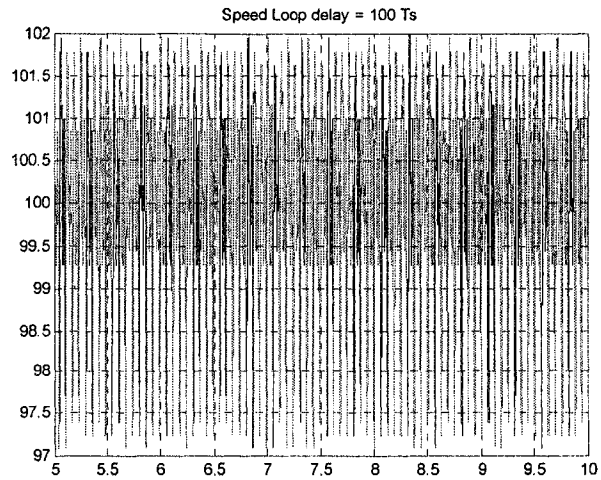


Fig. 4. 22 Experimental result for m=100.

Fi

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

4.8.2 Experimental Results

The complete diagram of the proposed self tuned NFC based speed ripple minimization of IMBRB drive system has been implemented in similar way as mentioned in Chapter 3. The photograph of the experimental system is shown in

Fig. **4.23**.

For comparison purpose, a PI-controller-based system is also developed and experimentally implemented. After trial and error, the proportional gain K_p and the integral gain K_i are selected as 0.4 and 0.3, respectively, so that the magnitudes of speed ripples can be comparable to the proposed NFC.

The sampling frequency is set to be 10 kHz in order to achieve a good tracking response. Due to the limitation of the computational power of the processor, in practice the proposed weights w_{1-9} and weights w_{19-27} are fixed based on simulation studies as following,

$$w_{1-9} = -3\ ,\ w_{19-27} = 3$$

Fig. 4.24 shows the zoom in view of the steady-state speed response for PI and proposed NFC based vector control of IMBRB drive. It is shown in this figure that proposed NFC and shows that the proposed NFC has eliminated the speed ripple up to 50%. Fig 4. 25 shows the FFT analytical results of speed response for PI controller and the proposed NFC, respectively. It is can be seen that the proposed NFC reduces the speed ripples with fundamental field frequency, second harmonic, fourth harmonic, and fifth harmonic.

Fig. 4.26 (a) - (c) show the changing trend of the weights w_{10-18} in a time segment of 100 seconds. It is evident that the proposed NFC is capable to minimize the fluctuation of speed and illustrate fault tolerant ability.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Fig. 4.23 Photograph of the laboratory experimental setup for IMBRB.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

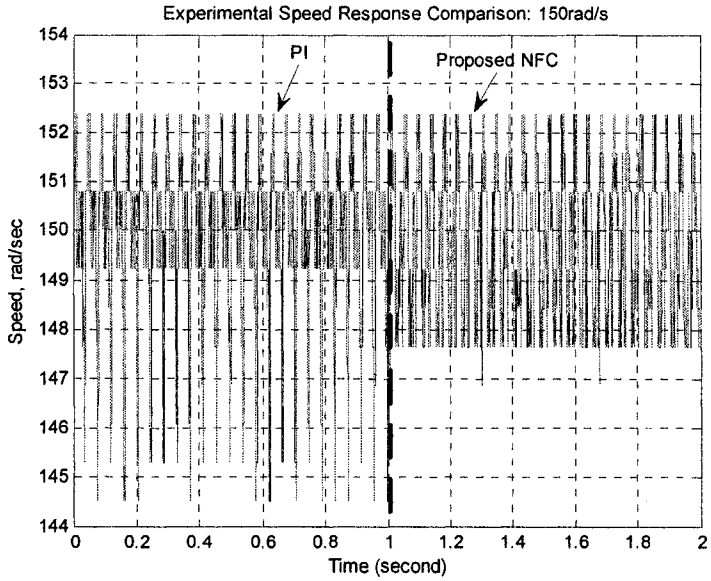


Fig. 4.24 Steady-state zoom-in-view of the experimental speed response at 150 rad/s.

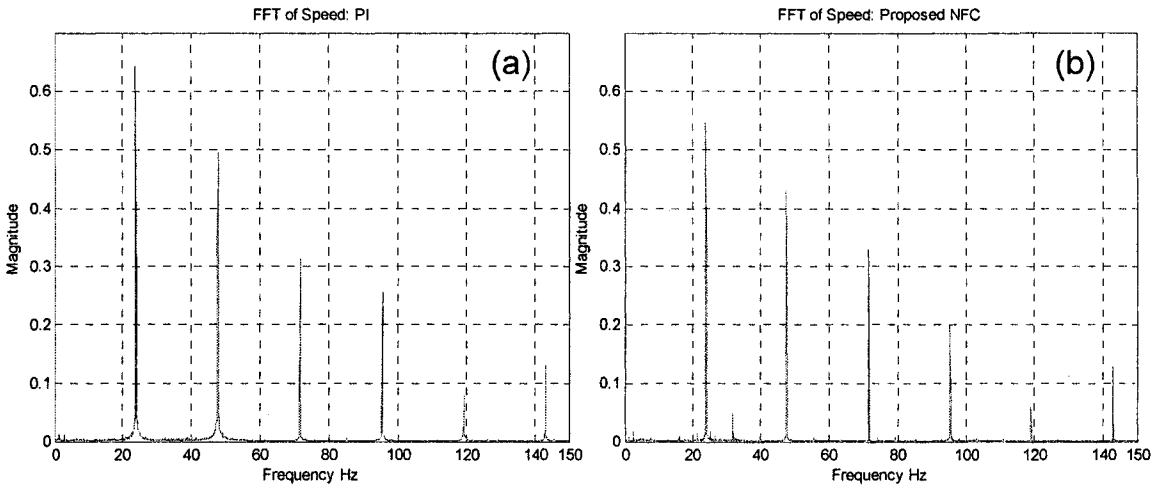


Fig 4. 25 FFT analysis of speed response for (a) PI, (b) the proposed NFC.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

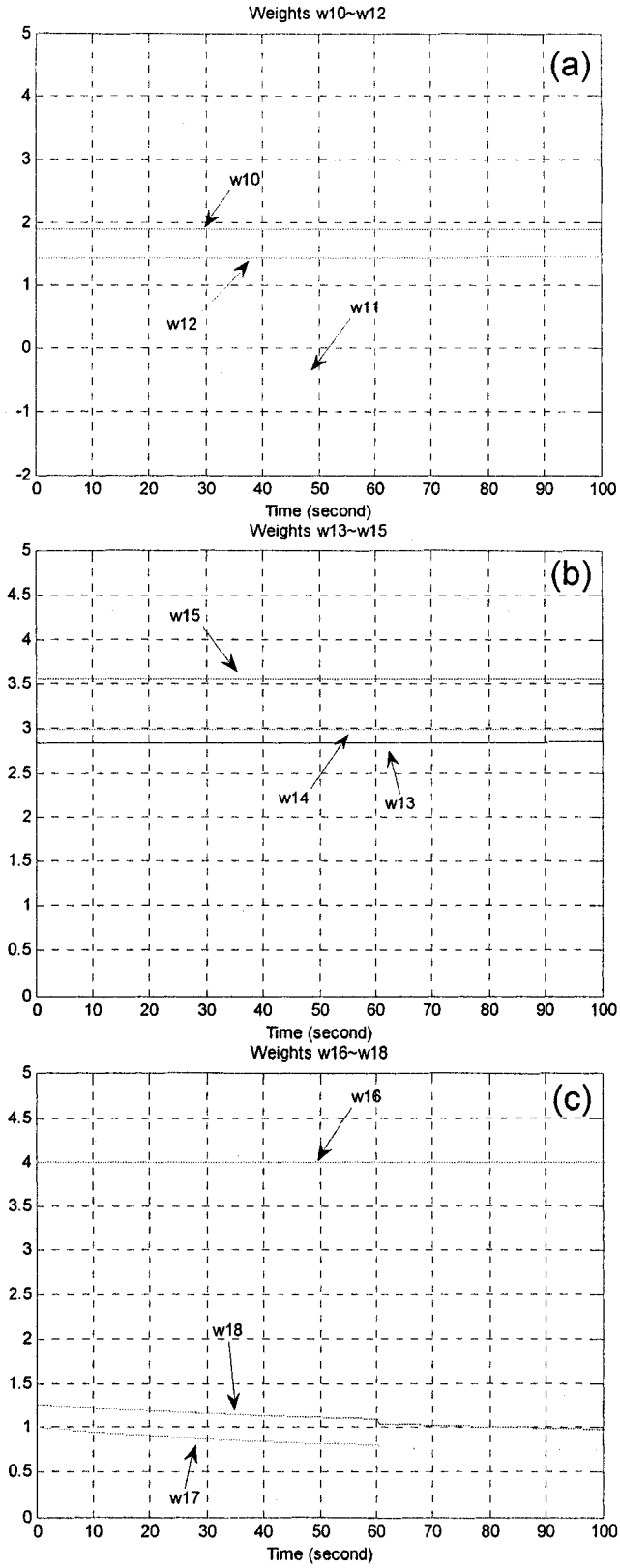


Fig. 4.26 Experimental results of weights variation at reference speed 150*rad/s*: (a)Weights 10-12; (b) Weights 13-15; (c) Weights 16-18.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

4.7 Conclusion

In this chapter, a novel NFC is developed to minimize the high frequency speed ripple of an IMBRB. A self tune algorithm has also been developed to update the weights according to the operating conditions. The convergence/divergence of the weights are investigated and validated by simulation results. A performance comparison with a PI controller showed that the proposed NFC can significantly improve the speed ripple. Thus, the proposed NFC acts like a fault tolerant controller for the IMBRB.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Chapter 5

Conclusion

This thesis develops an AIC for high performance IM drive in order to achieve fast and accurate speed response, quick recovery of speed from any disturbance such as parameter variations, load changes, step changes of command speed, etc. The intelligent controller has been developed in such a way that the computation burden is low and hence suitable for real-time application. Another intelligent controller has also been developed for an IMBRB in order to minimize the speed ripple and thus work as a type of fault tolerant controller. The particular work done in different chapters are briefly outlined below.

In Chapter 1, a state-of-the-art review of IM drives utilizing various control techniques has been provided. The motivation of the thesis and finally the organization of the thesis have also been presented.

In Chapter 2, the FOC theory and NFC with fuzzy singleton rules have been introduced. Various training algorithms such as BP, RLS and Kaczmarz's projection algorithms have been outlined. Inverse dynamics, differentiating model, and reinforcement learning have also been described as alternative solutions to the problem of training of NFC in a practical control system.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

In Chapter 3, a novel and low computational NFC have been developed for speed control of an IM drive. A supervised self-tuning method has been developed to adjust membership functions and weights of the developed NFC. The control of a three phase inverter using hysteresis current has been briefly discussed. The complete real-time implementation procedure utilizing the dSPACE DSP board DS1104 has also been provided.

In Chapter 4, the detail investigation about speed ripple for a faulty IMBRB has been provided. Another NFC has been developed for minimizing the speed ripple of an IMBRB. The performance of the NFC has tested both in simulation and experiment.

In a conclusion, the NFC has found to be a good candidate for high performance industrial drive applications.

5.1 Contributions

The contributions of this thesis are as follows,

- A novel, simplified NFC has been developed and successfully implemented in real-time for speed control of a high performance IM drive. Comparison with a conventional 2-input NFC showed that the proposed NFC does not decrease the system performance, while it improves the computational burden significantly.
- A new self-tuning algorithm has been developed to adjust the weights and membership functions for the simplified NFC. In this algorithm system error was directly used for self tuning the weights and membership functions. In this work the convergence/divergence of the weights has also been investigated and validated by simulation.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

- The speed ripple of IM and its relationship with rotor electrical angle have been investigated. A mathematical representation of speed ripple has been proposed and used to design the NFC.
- A novel NFC has been developed to minimize the speed ripple of an IMBRB. The developed technique has been tested both in simulation and experiment at different operating conditions. There is a close agreement between simulation and experimental results.

5.2 Future work

- From the experimental results, one can see that the developed NFC in Chapter 3 is still suffering from the overshoot and undershoot. This drawback could be solved by adaptively adjusting the tuning rates for the weights and membership functions.
- The objective of the work in Chapter 4 is to develop a FTC for the faulty IM. The design of a FTC involves two phases. The first phase is fault detection and isolation (FDI). In this phase the task is to design a dynamical system (filter) which, by processing input/output data, is able to detect the presence of an incipient fault and to isolate it from other faults and/or disturbance. Once the FDI filter has been designed, the second is to design a supervisory unit to reconfigure the controller so as to compensate for the effect of the fault and to fulfill performance constraint. In Chapter 4, instead of following the conventional procedure to develop a FTC, a NFC has been developed to minimize speed ripple and act as a FTC. In future, the FDI for an IMBRB and a FTC for IMBRB utilizing the procedure mentioned earlier can be developed.
- When NFC is trained for a particular work, a large training rate would lead to worse performance or even instability. In the future, the analysis of the stability and robustness could be done to find a suitable tuning rate.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

References

[1] Giuseppe S. Buja, Marian P. Kazmierkowski, "Direct Torque Control of PWM Inverter-Fed AC Motors—A Survey", *IEEE Transactions on industrial electronics*, Vol.51, No. 4, August 2004.

[2] D. W. Novotny, T. A. Lipo, "Vector Control and Dynamics of AC Drives", *Oxford*, 1996.

[3] Andrzej M. Trzynadlowski, "Field Orientation Principle in Control of Induction Motors", *Kluwer Axsdemic*, 1994.

[4] R. dc Doncker, D.W. Novotny, "The universal field oriented controller", *IEEE-IAS Annual Meeting Conference Record*, October 1988, pp. 450-456.

[5] I. Takahashi, N. Noguchi, "A new quick response and high efficiency control strategy of an induction motor", *IEEE Transactions on Industry Application*, Vol. 22, No. 5, Sept./Oct. 1986, pp. 820-827.

[6] M. Depenbrok, "Direct Self-Control (DSC) of inverter-fed induction machine", *IEEE Transactions on Power Electronics*, Vol. PE-3, No. 4 Oct. 1988, pp. 420-429.

[7] T.G. Habetler, D.J. Divan, "Control Strategies for Direct Torque Control Using Discrete Pulse Modulation", *IEEE Transactions on Industry Applications*, Vol. 27. No. 5 Sept. /Oct. 1991, pp.893-901.

[8] T.G. Habetler, F. Profumo, M. Pastorelli, L.M. Tolbert, "Direct Torque Control of Induction Machines Using Space Vector Modulation", *IEEE Transactions on industry Applications*, Vol. 28, No. 5 Sept./Oct. 1992, pp. 1045-1053.

[9] G. Buja, D. CGadei, G. Serra, "DTC-based strategies for induction motor drives", *Conference Preceding of IECON'97*, pp. 1506-1516.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

[10] R. Marino, "Output feedback control of current-fed induction motors with unknown rotor resistance", *IEEE Transactions Control System Technology*, Vol.4, July 1996, pp. 336–347.

[11] M. P. Kazmierkowski, D. L. Sobczuk, "Sliding mode feedback linearizedcontrol of PWM inverter-fed induction motor", *IEEE IECON'96*, Taipei, Taiwan, R.O.C., 1996, pp. 244–249.

[12] M. P. Kazmierkowski, R. Krishnan, F. Blaabjerg, "Control in Power Electronics", *San Diego, CA: Academic*, 2002.

[13] M. Pietrzak-David, B. de Fornel, "Non-Linear control with adaptive observer for sensorless induction motor speed drives", *EPE Journal*, Vol. 11, No. 4, 2001, pp. 7–12.

[14] R. Ortega, A. Loria, P. J. Nicklasson, H. Sira-Ramirez, "Passivity Based Control of Euler-Lagrange Systems", *London, U.K.: Springer-Verlag*, 1998.

[15] Z. Krzeminski, "Nonlinear control of induction motors", *Processing 10th IFAC World Congr., Munich, Germany*, 1987, pp. 349–354.

[16] P. C. Krause, C. H. Thomas, "Simulation of symmetrical induction machinery", *IEEE Transaction on Power Apparatus System*, Vol. PAS-84, No. 11, 1965, pp.1038-1053.

[17] Hoang Le-Huy, "Comparison of Field-Oriented Control and Direct Torque Control for Induction Motor Drives", *Conference Record - IAS Annual Meeting*, Vol. 2, 1999, pp. 1245-1252.

[18] Gerardo Espinosa-Perez, Romeo Ortega, "Tuning of PI gains for FOC of induction motors with guaranteed stability", *IECON Proceedings*, Vol. 2, 1997, pp. 569-574.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

[19] G. Espinosa-Perez, G.W. Chang, R. Ortega, E. Mendes, "On field-oriented control of induction motors: Tuning of the PI gains for performance enhancement", *Proceedings of the IEEE Conference on Decision and Control*, Vol. 1, 1998, pp. 971-976.

[20] Eun-Chul. Shin, Tae-Sik. Park, Won-Hyun Oh, Ji-Yoon Yoo, "A Design Method of PI Controller for an Induction Motor with Parameter Variation", *IECON Proceedings*, Vol. 1, 2003, pp. 408-413.

[21] Gi-Won Chang, Gerardo Espinosa-Perez, Eduardo Mendes, Romeo Ortega, "Tuning rules for the PI gains of field-oriented controllers of induction motors", *IEEE Transactions on Industrial Electronics*, Vol. 47, No. 3, June, 2000, pp. 592-602.

[22] Lin Feng, Zheng Hongtao, Yang Qiwen, "Sensorless vector control of induction motors based on online GA tuning PI controllers", *Fifth International Conference on Power Electronics and Drive Systems*, 2003, pt. 1,Vol.1, pp. 222-225.

[23] Sazali Yaacob, Faisal A. Mohamed, "Real time self tuning controller for induction motor based on PI method", *Proceedings of the SICE Annual Conference*, 1999, pp. 909-914.

[24] R. D. Lorenz, D. B. Lawson, "A simplified approach to continuous, on-line tuning of field-oriented induction machine drives'', *IEEE-IAS Conference Record. (Pittsburgh)*, 1988, pp. 444-449.

[25] Timothy M. Rowan, Russel J. Kerkman, David Leggate, "A simple on-line adaption for Indirect Field Orientation of an induction machine", *IEEE Transactions on industry applications*, Vol. 27, No. 4, July-August 1991.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

[26] E. Y. Y. Ho, P. C. Sen, "Control dynamics of speed drive systems using sliding mode controllers with integral compensation", *IEEE Transactions on industry applications*, Vol. 27, Sept./Oct. 1991, pp. 883–892.

[27] K. K. Shyu, H. J. Shieh, "A new switching surface sliding mode speed control for induction motor drive systems", *IEEE Transactions on Power Electronics*, Vol. 11, July 1996, pp. 660–667.

[28] Z. Hakan Akpolat, Greg M.Asher, Jon C Clare, "A Practical approach to the design of robust speed controllers for machine drives", *IEEE Transactions on Industrial Electronics*, Vol. 47, No. 2, Apr, 2000, pp. 315-324.

[29] J. J. E. Slotine, W. Li, "Applied Nonlinear Control", *Englewood Cliffs, NJ: Prentice-Hall*, 1991.

[30] Hualin Tan, Jie Chang, "Field orientation and adaptive backstepping for induction motor control", *Conference Record - IAS Annual Meeting*, Vol. 4, 1999, pp. 2357-2363.

[31] C. Attaianese, G. Tomasso, "H infinity control of induction motor drives", *IEEE Proceedings: Electric Power Applications*, Vol. 148, No. 3, May, 2001, p272-278.

[32] Stefano Chiaverini, Gennaro Figalli, Giuseppe Fusco, "H infinity design of a robust speed controller for induction motors", *IEEE Conference on Control Applications - Proceedings*, Vol. 2, 1999, pp. 956-961.

[33] Giuseppe Fusco, Stefano Chiaverini, "Bandwidth vs. gains design of H infinity tracking controllers for current-fed induction motors", *Automatica*, Vol. 38, No. 9, September 2002, pp. 1575-1581.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

[34] C.P. Bottura, S.A. Filho, J.L Silvino, "H2 robust digital vector control for the induction machine", *IEEE Proceedings: Control Theory and Applications*, Vol. 143, No. 3, May 1996, pp. 237-243.

[35] Celso Pascoli Bottura, Manoel Francisco Soares Neto, Sergio Antonio Augusto Filho, "Robust speed control of an induction motor: An H infinity control theory approach with field orientation and μ-analysis", *IEEE Transactions on Power Electronics*, Vol. 15, No. 5, September 2000, pp. 908-915.

[36] A. Makouf, M.E.H. Benbouzid, D. Diallo, N.E Bouguechal, "Induction motor robust control: An H infinity control approach with field orientation and input-output linearizing", *IECON Proceedings*, Vol. 2, 2001, pp. 1406-1411.

[37] Michael T. Wishart, Ronald G Harley, "Identification and control of induction machines using artificial neural networks", *IEEE Transactions on Industry Applications*, Vol. 31, No. 3, May-June 1995, pp. 612-619.

[38] A. Ba-razzouk, A. Cheriti, G. Olivier, "Neural networks based field oriented control scheme for induction motors", *Conference Record - IAS Annual Meeting*, Vol. 2, 1997, pp. 804-811.

[39] H.A.F Mohamed, W.P. Hew, "A neural network vector control of induction motor", *2000 TENCON Proceedings*, 2000, pt. 3, pp. 336-41.

[40] Bruce Burton, Farrukh Kamran, Ronald G. Harley, Thomas G Habetler, Martin Brooke, Ravi Poddar, "Identification and control of induction motor stator currents using fast on-line random training of a neural network", *Conference Record - IAS Annual Meeting*, Vol. 2, 1995, pp. 1781-1787.

[41] K.L. Shi, T.F. Chan, Y.K. Wong, "Direct self control of induction motor based on neural network", *IEEE Transactions on Industry Applications*, Vol. 37, No. 5, September/October, 2001, pp. 1290-1298.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

[42] Tien-Chi Chen, Tsong-Terng Sheu, "Model reference neural network controller for induction motor speed control", *IEEE Transactions on Energy Conversion*, Vol. 17, No. 2, June 2002, pp. 157-163.

[43] M.N. Uddin, T.S. Radwan, M.A. Rahman, "Performances of fuzzy-logic-based indirect vector control for induction motor drive", *IEEE Transactions on Industry Applications*, Vol. 38, No. 5, September/October 2002, pp. 1219-1225.

[44] T. C. Huang, M. A. El-Sharkawi, "High performance speed and position tracking of induction motors using multi-layer fuzzy control", *IEEE Transactions on Energy Conversion*, Vol. 11, No. 2, June 1996, pp. 353-358.

[45] S. Bolognani, M. Zigliotto, "Hardware and software effective configurations for multi-input fuzzy logic controllers", *IEEE Transactions on Fuzzy System*, Vol. 6, February 1998, pp. 173–179.

[46] Y. Tang, L. Xu, "Fuzzy logic application for intelligent control of a variable speed drive", *IEEE Transaction on Energy Conversion*, Vol. 9, December 1994, pp.679–685.

[47] C. H. Won, S. Kim, B. K. Bose, "Robust position control of induction motor using fuzzy logic control", *IEEE IAS Conference Record, Houston, TX*, 1992, pp. 472–481.

[48] F. F. Cheng, S. N. Yeh, "Application of fuzzy logic in the speed control of AC servo system and an intelligent inverter", *IEEE Transactions on Energy Conversion*, Vol. 8, No. 2, June 1993, pp. 312-18.

[49] Xiangjie Liu, Tianyou Chai, "A Class of neural-fuzzy controller and its application in AC speed adjustment with varying frequency", *Proceedings of the IEEE Conference on Decision and Control*, Vol. 3, 1997, pp. 2742-2747.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

[50] M. N. Uddin, H. Wen, "Development of a self-tuned neuro-fuzzy controller for induction motor drives", *IEEE Transactions on Industry Applications*, Vol. 43, No. 4, July/August 2007, pp. 1108 - 1116.

[51] Sang Hoon Kim, Lark Kyo Kim, "Design of a neuro-fuzzy controller for speed control applied to AC servo motor", *IEEE International Symposium on Industrial Electronics*, Vol. 1, 2001, pp. 435-440.

[52] Hao Wen, M.N. Uddin, "Filter based torque ripple minimization of an adaptive neuro-fuzzy controller for induction motor", *2005 Canadian Conference on Electrical and Computer Engineering*, 2006, pp. 474-477.

[53] A. Vahedi, F. Rashidi, Hosseinazizi, "Sensorless speed control of induction motor drives using a robust and adaptive neuro-fuzzy based intelligent controller", *2004 IEEE International Conference on Industrial Technology*, 2004, pt. 2, Vol. 2, pp. 617-27.

[54] M. Gokbulut, C. Bal, B. Dandil, "A virtual electrical drive control laboratory: neuro-fuzzy control of induction motors", *Computer Applications in Engineering Education*, Vol. 14, No. 3, 2006, pp. 211-21.

[55] G. El-Saady, A.M. Sharaf, A.M. Makky, M.K. El-Sherbiny, G. Mohamed, "An error driven hybrid neuro-fuzzy torque/speed controller for electrical vehicle induction motor drive", *Proceedings of the Intelligent Vehicles '94 Symposium*, 1994, pp. 449-454.

[56] S. Weerasooriya, M.A. El-Sharkawi, "Laboratory implementation of a neural network trajectory controller for a DC motor", *IEEE Transactions on Energy Conversion*, Vol. 8, No. 1, March 1993, pp. 107 – 113.

[57] Faa-Jeng Lin, Rong-Jong Wai, Hong-Pong Chen, "A PM synchronous servo motor drive with an on-line trained fuzzy neural network controller", *IEEE*

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Transactions on Energy Conversion, Vol. 13, No. 4, December 1998, pp. 319 – 325.

[58] P. Grabowski, F. Blaabjerg, "Direct torque neuro-fuzzy control of induction motor drive, DSP implementation", *IEEE IECON Proceedings*, 1998, pp. 657-662.

[59] F-J Lin, R-J Wai, "Adaptive fuzzy-neural-network control for induction spindle motor drive", *IEEE Transactions on Energy Conversion*, Vol. 17, No. 4, December 2002, pp. 507–513.

[60] S. Mir, M.S. Islam, T. Sebastian, I. Husain, "Fault-tolerant switched reluctance motor drive using adaptive fuzzy logic controller", *IEMDC'03. IEEE International Electric Machines and Drives Conference*, 2003, pt. 2, vol.2, pp. 835-841.

[61] R. Anita, B. Viswanathan, B. Umamaheswari, "Implementation of neuro fuzzy fault tolerant sliding mode controller for a DC servo motor", *IETE Technical Review*, Vol. 18, No. 5, September/October 2001, pp. 415-420.

[62] Pierre Yves Glorennec, "Neuro-fuzzy control using reinforcement learning", *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Vol. 4, 1993, pp.91-96.

[63] A. Jihen, I Slama-Belkhodja, "Investigation of fault tolerant of direct torque control in induction motor drive", *IEEE Conference Publication, Vol. 1, Second International Conference on Power Electronics, Machines and Drives, PEMD 2004*, 2004, pp. 67-72.

[64] K.-S. Lee, J.-S. Ryu, "Instrument fault detection and compensation scheme for direct torque controlled induction motor drives", *IEEE Proceedings: Control Theory and Applications*, Vol. 150, No. 4, July 2003, pp. 376-382.

[65] M.B. De Rossiter Correa, C.B. Jacobina, E.R. Cabral Da Silva, A.M. Nogueira Lima, "An induction motor drive system with improved fault tolerance", *IEEE*

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Transactions on Industry Applications, Vol. 37, No. 3, May/June 2001, pp. 873-879.

[66] Claudio Bonivento, Aliberto Isidori, Lorenzo Marconi, Andrea Paoli, "Implicit fault-tolerant control: Application to induction motors", *Automatica*, Vol. 40, No. 3, March 2004, pp. 355-371.

[67] A. Rubaai , D. Rickattes, M. D. Kankam, " Development and Implementation of an Adaptive Fuzzy-Neural Network Controller for Brushless Drives", *IEEE Transactions on Industry Applications*, Vol. 38, No.2, March/April 2002, pp. 441-447.

[68] P. Grabowski, F. Blaabjerg, "Direct torque neuro-fuzzy control of induction motor drive, DSP implementation", *IEEE IECON Proceedings*, 1998, pp. 657-662.

[69] S.J. Ovaska, S. Valiviita, "Angular acceleration measurement: a review", *IEEE Transactions on Instrumentation and Measurement*, Vol. 47, No. 5, October 1998, pp. 1211-17.

[70] P.B. Schmidt, R.D. Lorenz, "Design principles and implementation of acceleration feedback to improve performance of dc drives", *IEEE Transactions on Industry Applications*, Vol.28, No.3, May/June 1992, pp.594-599.

[71] S. Valiviita, S.J. Ovaska, "Delayless recursive differentiator with efficient noise attenuation for motion control applications", *IECON Proceedings*, Vol. 3, 1998, pp. 1481-1486.

[72] "Matlab, Simulink User Guide", *Math Works Inc.*, 2003.

[73] "dSPACE Implementation Guide", *Paderborn, Germany*, 2003.

[74] C. T. Lin, C. S. G. Lee, "Neural fuzzy systems: a neuro-fuzzy synergism to intelligent systems", Prentice Hall, 1996.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

[75] M. Nasir Uddin, Z. Huang and M. I. Chy, "A Simplified Self-Tuned Neuro-Fuzzy Controller Based Speed Control of an Induction Motor Drive", *IEEE Power Engineering Society Annual Conference and Meeting (PES-2007), USA, 2007.*

[76] K. P. Venugopal, R. Sudhakar, A. S. Pandya, "An Improved Scheme for Direct Adaptive Control of Dynamical Systems using Back Propagation Neural Networks", *Circuits System Signal Process*, Vol.14, No. 2, 1995, pp. 213-236.

[77] Jhy-Shing Roger Jang, Chuen-Tsai Sun, Eiji Mizutani, "Neuro-Fuzzy and Soft Computing: Computational Approach to Learning and Machine Intelligence", *Prentice Hall*, 1997.

[78] A. B. Sasi, B. Payne, A. York, F. Gu, A. Ball. "Condition Monitoring of Electric Motors Using Instantaneous Angular Speed", *paper presented at the Maintenance and Reliability Conference (MARCON)*, Gatlinburg, TN.

[79] S. J. Chapman, "Electric Machinerv Fundamentals", *New York: McGraw-Hill, Inc.*, 1991.

[80] B. K. Bose, "Power Electronics & AC Drives", *Englewood Cliffs, NJ: Prentice Hall International*, 1986.

[81] Dae-Woong Chung, Seung-Ki SUI, Dong-Choon Lee, "Analysis and Compensation of Current Measurement Error in Vector Controlled AC Motor Drives", *IEEE IAS Annual Meeting*, 1996, pp. 388-393.

[82] M.S. Khanniche, Yi Feng Guo, "A microcontroller-based real-time speed measurement for motor drive systems", *Journal of Microcomputer Applications*, Vol. 18, No. 1, January 1995, pp. 39-53.

[83] F. Briz, J.A. Cancelas, A. Diez, "Speed measurement using rotary encoders for high performance ac drives", *IECON Proceedings* , Vol. 1, 1994, pp. 538-542.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

[84] A.H. Kadhim, T.K.M. Babu, D. O'Kelly, "Measurement of steady-state and transient load-angle, angular velocity, and acceleration using an optical encoder", *IEEE Transactions on Instrumentation and Measurement*, Vol. 41, No. 4, August 1992, pp. 486-489.

[85] John N. Lygouras, Konstantinos A. Lalakos, Phillipos G. Tsalides, "High-performance position detection and velocity adaptive measurement for closed-loop position control", *IEEE Transactions on Instrumentation and Measurement*, Vol. 47, No. 4, August 1998, pp 978-985.

[86] Ronald H. Brown, Susan C. Schneider, Michael G. Mulligan, "Analysis of algorithms for velocity estimation from discrete position versus time data", *IEEE Transactions on Industrial Electronics*, Vol. 39, No. 1, February 1992, pp. 11-19.

[87] Phillips, Stephen M., Branicky, Michael S. "Velocity Estimation Using Quantized Measurements", *Proceedings of the IEEE Conference on Decision and Control*, Vol. 5, 2003, pp. 4847-4852.

[88] B. Liang, A.D. Ball, S.D. Iwnicki, "Simulation and fault detection of three-phase induction motors", *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, Vol. 3, 2002, pp. 1813-1817.

[89] Jong-Woo Choi, Sang-Sup Lee, Sang-Yeop Yu, Seok-Joo Jang, "Novel periodic torque ripple compensation scheme in vector controlled AC motor drives", *Conference Proceedings - IEEE Applied Power Electronics Conference and Exposition - APEC*, Vol. 1, 1998, pp. 81-85.

[90] Roberto Barro, Ping. Hsu, "Torque ripple compensation of vector controlled induction machines", *PESC Record - IEEE Annual Power Electronics Specialists Conference*, Vol. 2, 1997, pp. 1281-1287.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

[91] Karl Johan Åström, Björn Wittenmark, "Adaptive Control", *Addison-Wesley*, 1995.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Appendix – A

IM parameters

1. Parameters of IM (used in Chapter 3)

Power	$500W$
Stator resistance	4.12Ω
Rotor resistance	1.797Ω
Number of pole pairs	1
Stator inductance	$0.0114H$
Rotor inductance	$0.0073H$
Mutual inductance	$0.245H$
Inertia	$0.000927kgm^2$
Rated speed	$3600RPM$

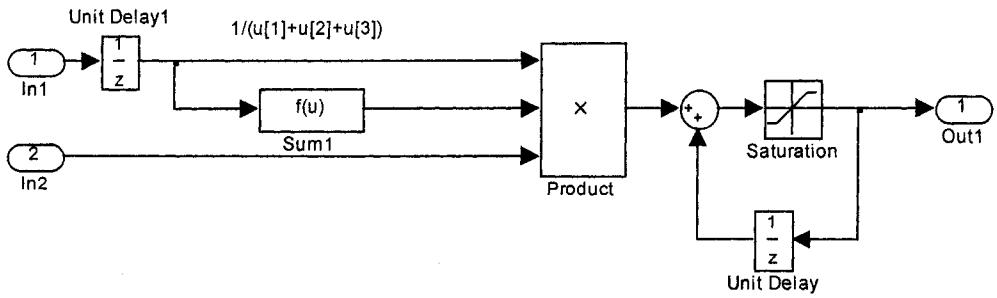
2. Parameters of IMBRB (used in Chapter 4)

Power	$250W$
Stator resistance	6.5Ω
Rotor resistance	3.4Ω
Number of pole pairs	2
Stator inductance	$0.0103H$
Rotor inductance	$0.0154H$
Mutual inductance	$0.2655H$
Inertia	$0.0012kgm^2$
Rated speed	$1725RPM$

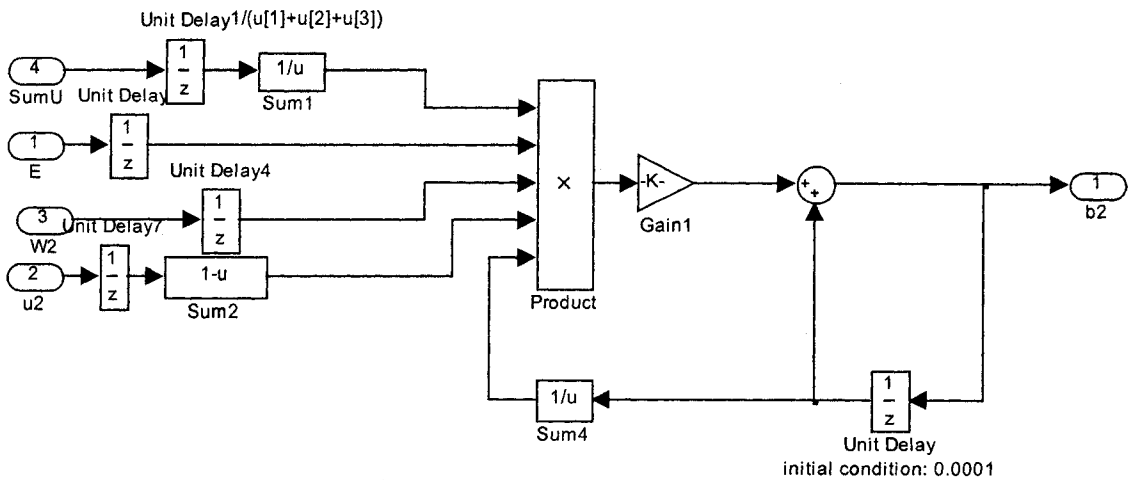
Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Appendix – B

Simulink Subsystem for Simplified NFC (Fig. 3.7)

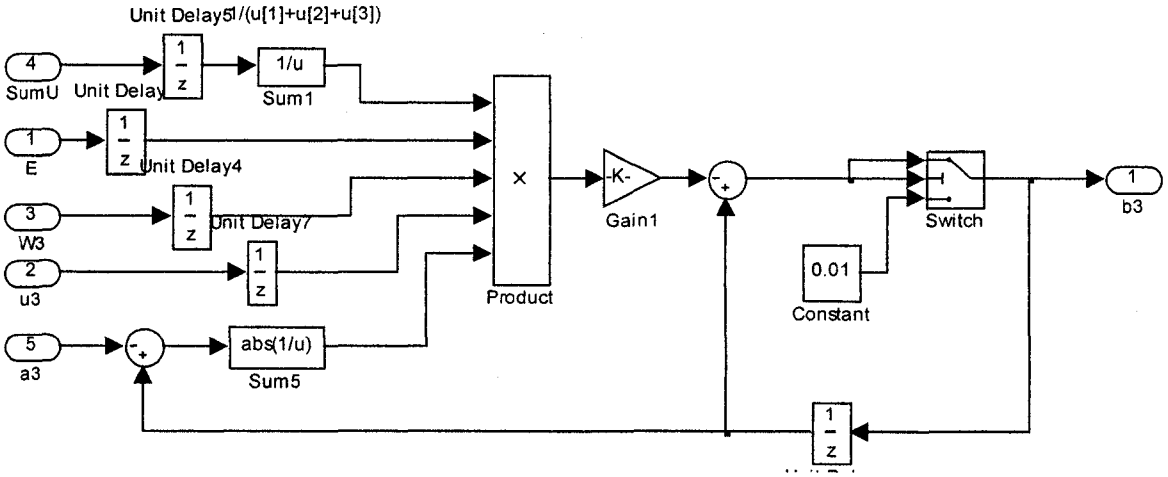


B1. Subsystem of "WeightsTuning".

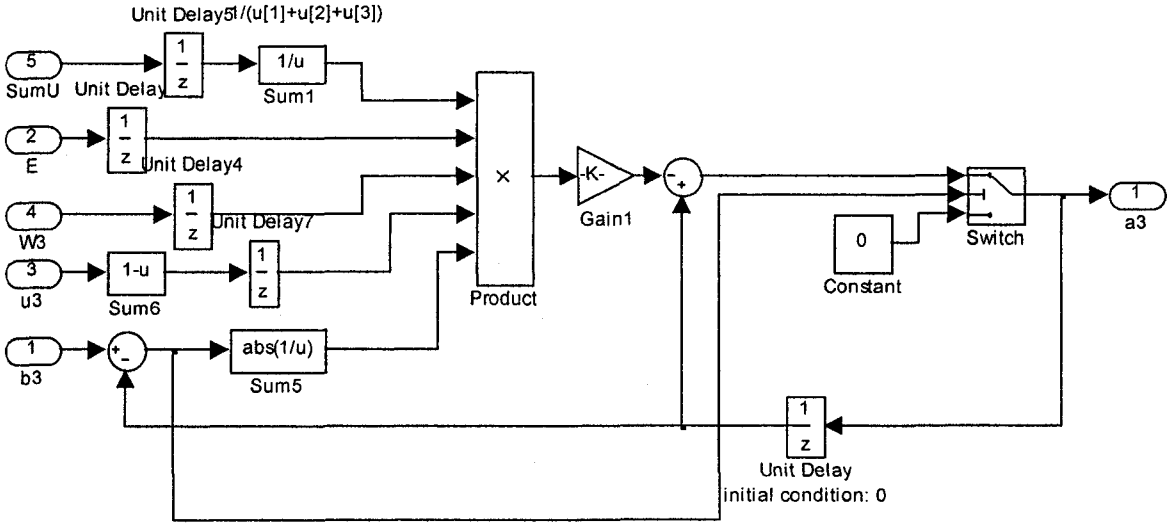


B2. Subsystem of "MemTuning".

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

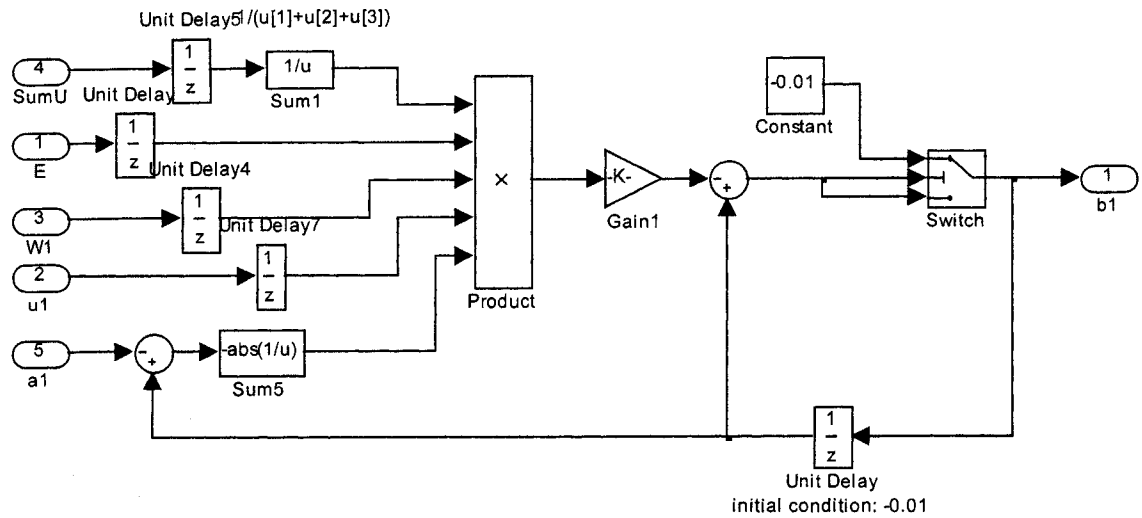


B3. Subsystem of "MemTuning1".

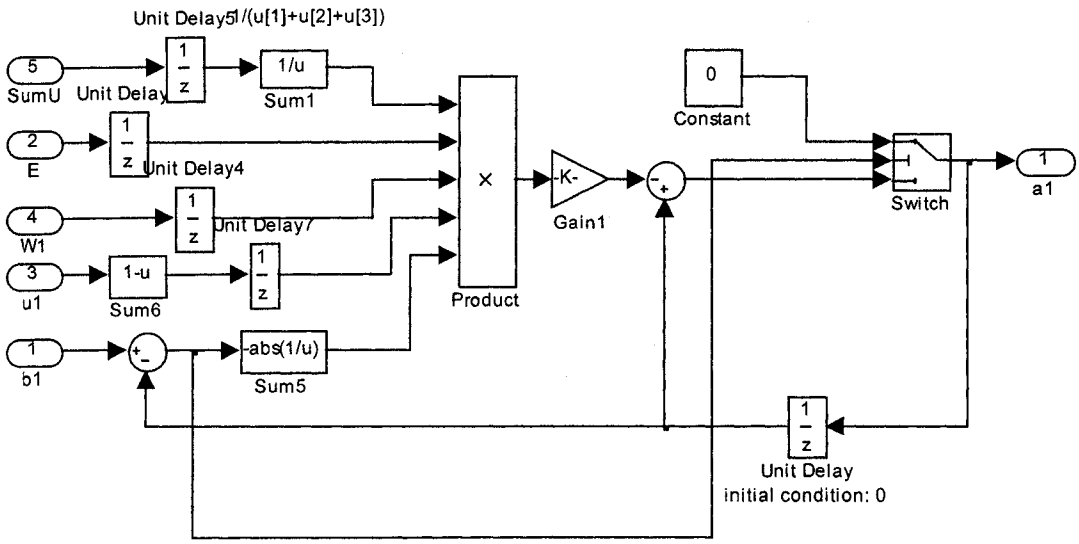


B4. Subsystem of "MemTuning2".

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.



B5. Subsystem of "MemTuning3".

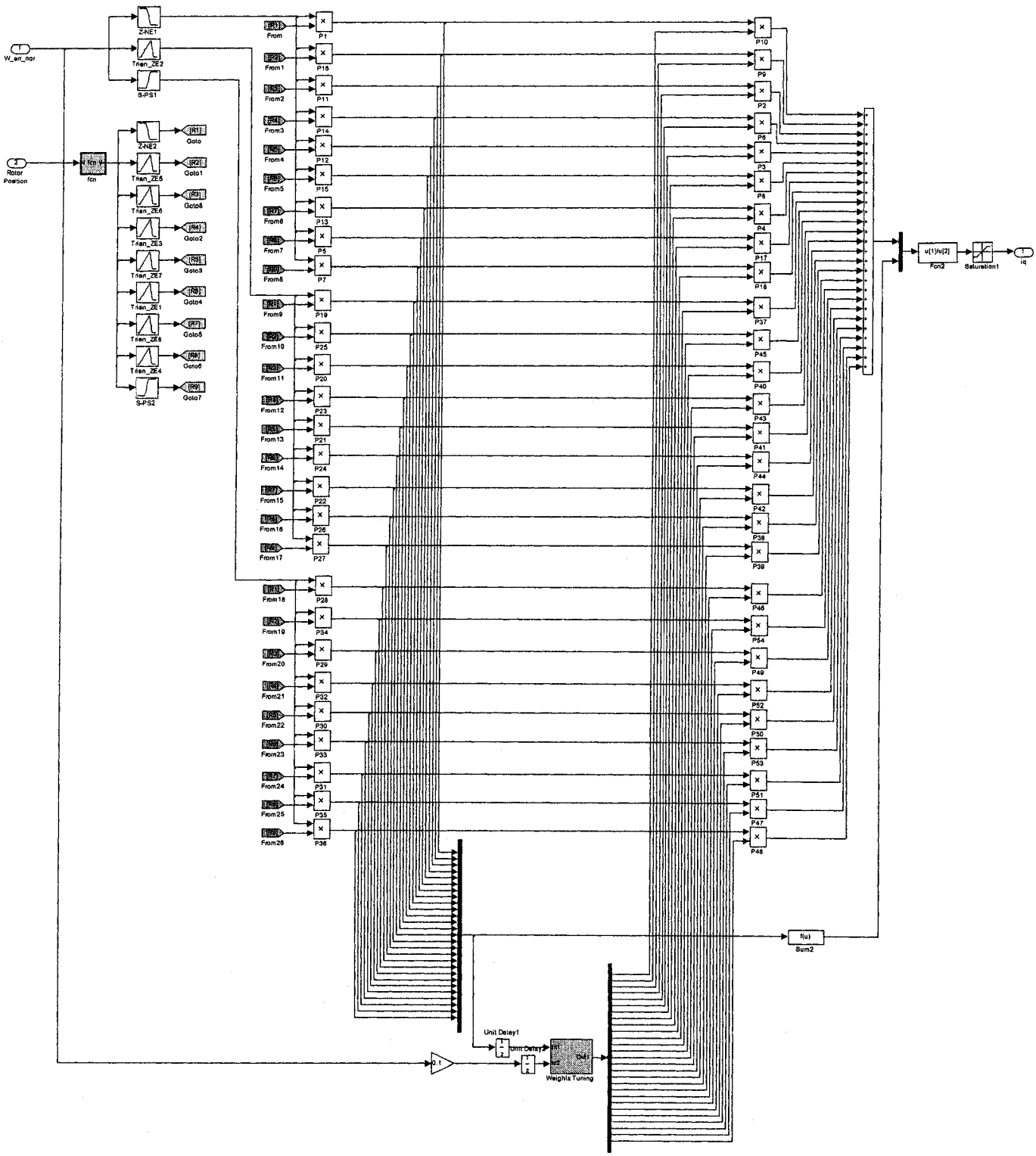


B6. Subsystem of "MemTuning4".

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

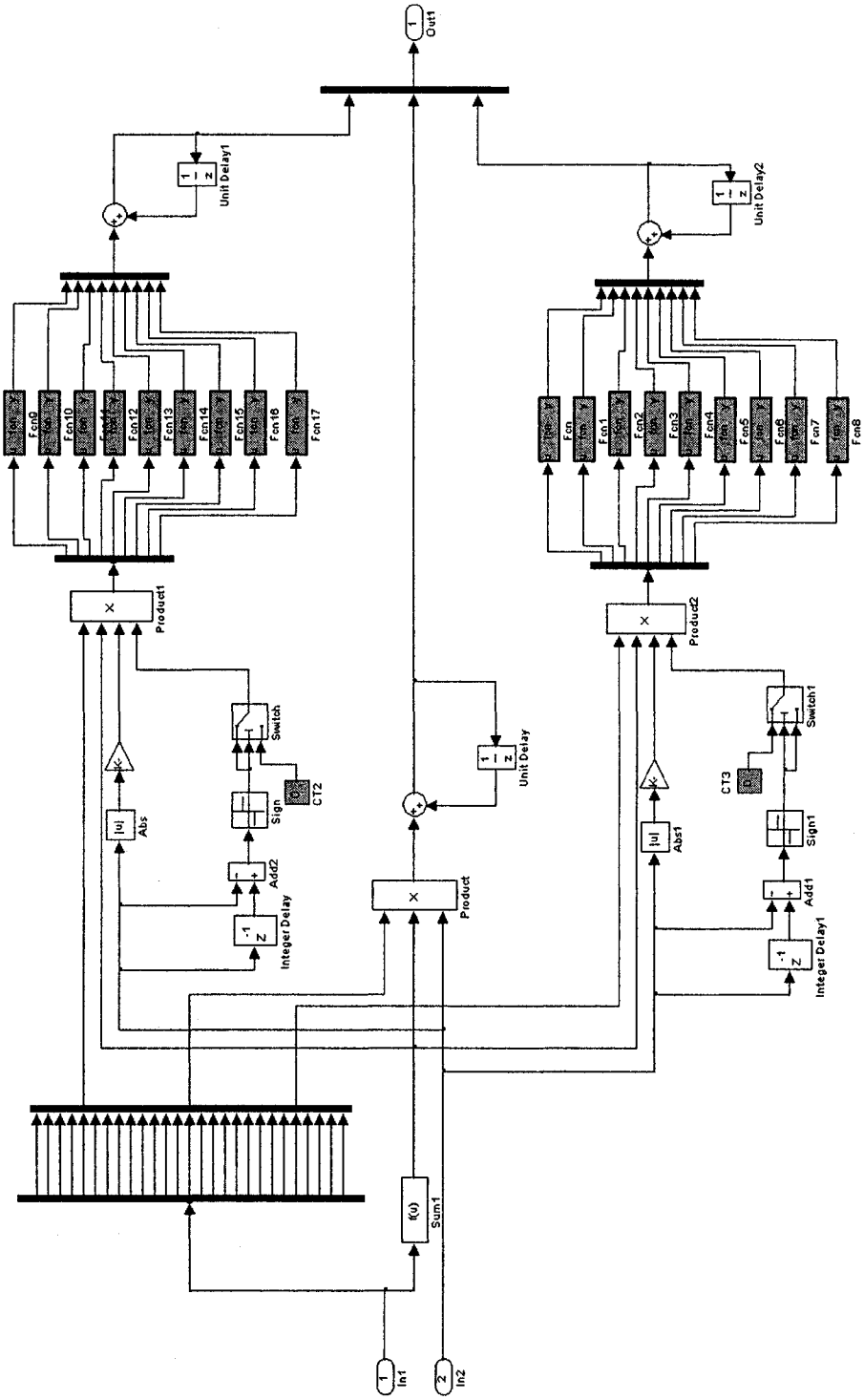
Appendix – C

Simulink Model of NFC for IMBRM Drive



C1. Simulink model of NFC developed for IMBRB drive (Chapter 4)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

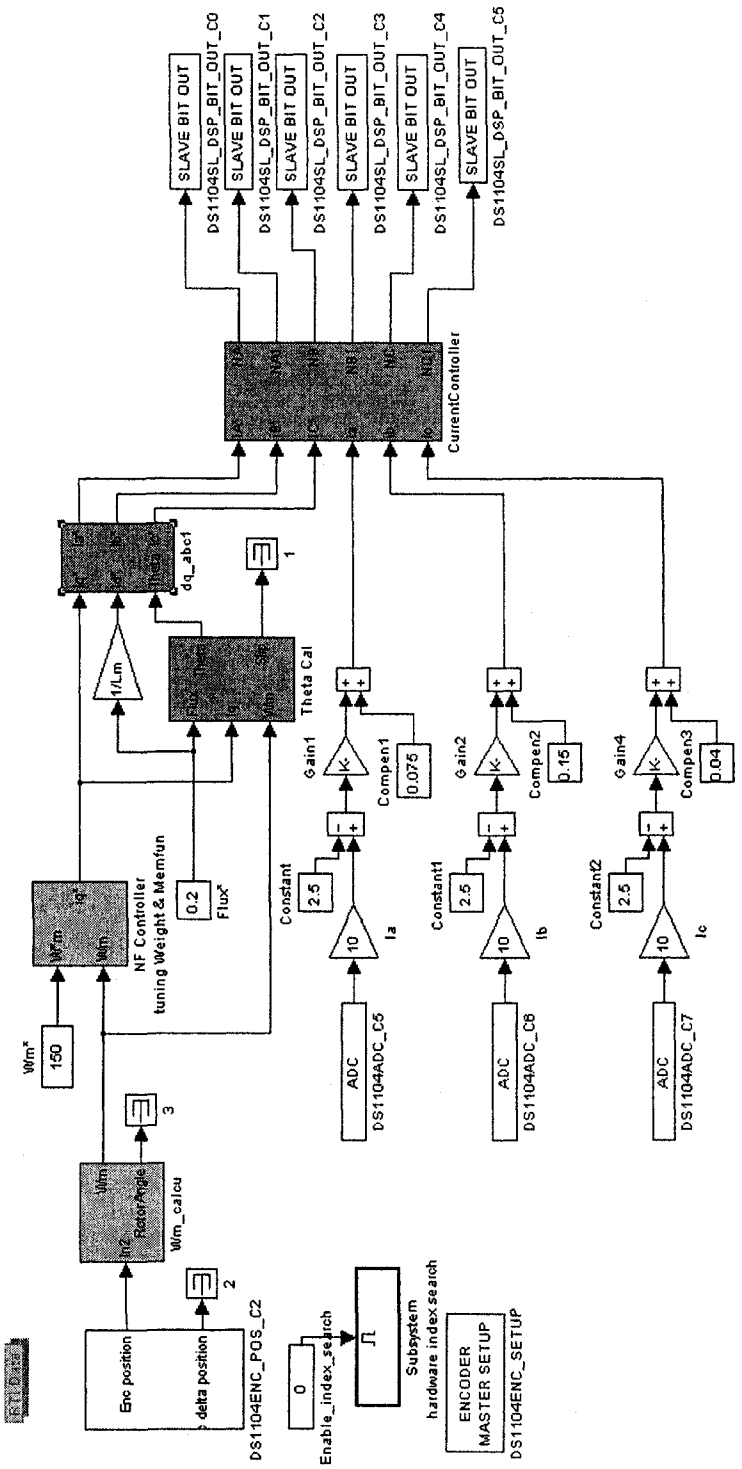


C2. Subsystem of "Weights Tuning" in Fig. C.1

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

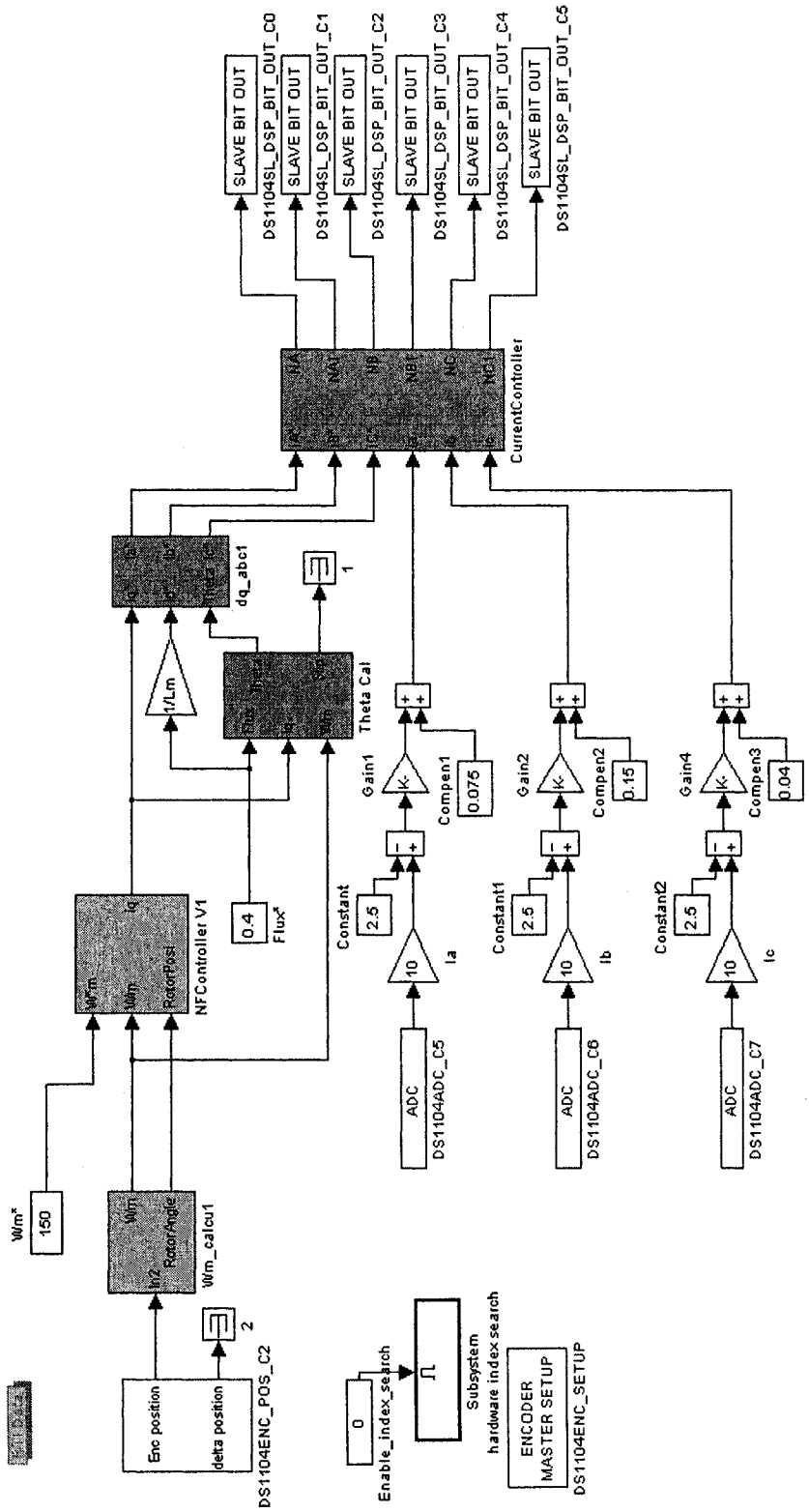
Appendix – D

Real-time Simulink Model



D1. Real-time Simulink model of low computational NFC based IM control scheme (Chapter 3)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.



D2. Real-time Simulink model of NFC based speed minimization of IMBRB control scheme (Chapter 4)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.